

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістра

(назва освітнього ступеня)

на тему: Розробка мобільного додатку на базі Android для людей з інклюзією

Виконав(ла): студент(ка) 6 курсу, групи СПм-61
спеціальності 121-Інженерія програмного забезпечення

(шифр і назва спеціальності)

Мінько В.В.
(підпис) (прізвище та ініціали)

Керівник Дячук С.Ф.
(підпис) (прізвище та ініціали)

Нормоконтроль
(підпис) (прізвище та ініціали)

Завідувач кафедри Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Тернопіль 2021

РЕФЕРАТ

Кваліфікаційна робота на здобуття ступеня магістра на тему “Розробка мобільного додатку на базі Android для людей з інклюзією”, Мінька Віталія Вікторовича. Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних система і програмної інженерії, Кафедра програмної інженерії, група СПм-61, Тернопіль 2021. С. – , рис – , табл. – , додат. – , бібліогр. – .

Актуальність обраної теми зумовлена унікальною ситуацією, яка склалася сьогодні як в Україні так і у світі. Сьогодні кожен із нас може себе відчувати на місці людини, з обмеженими можливостями. Метою роботи є аналіз, моделювання, проектування та розробка програмного забезпечення для з використанням норм доступності для людей з інклюзією для полегшення та покращення їх життєздатності. Результатом виконання роботи є додаток на базі Android для людей з інклюзією. **Ключові слова:** ІНКЛЮЗІЯ, ДОСТУПНІСТЬ, МОЖЛИВОСТІ, ОПЕРАЦІЙНА СИСТЕМА, ANDROID, ОБМЕЖЕННЯ.

ABSTRACT

Qualifying work for a master's degree “Development of mobile application with using android for inclusive people”. Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Software Engineering Department, group SPm-61, Ternopil, 2021. C. - , Fig. – , Table. - . Add. - , Ref. - .

The relevance of the chosen topic is due to the unique situation that has developed today in Ukraine and in the world. Today, each of us can feel in the place of a person with disabilities. The aim of the work is to analyze, model, design and develop software using accessibility standards for people with inclusion to facilitate and improve their viability. The result is an Android-based app for people with inclusion. **Keywords:**

INCLUSION, ACCESSIBILITY, CAPABILITIES, OPERATING SYSTEM, LIMITATIONS.

ЗМІСТ

Вступ.....	
1. Аналітична частина.....	
1.1 Аналіз доступності.....	
1.2 Аналіз області доступності програмних систем у світі	
1.3 Демонстрація доступності у програмних системах.....	
2. Цифрове включення інклюзивності	
2.1 Універсальний дизайн	
2.2 Інклюзивний дизайн	
2.3 Цифрове включення.....	
2.4 Аналіз профілів людей для визначення цифрового включення	
2.5 Інструкції доступності контенту для мобільних пристроїв.....	
2.6 Правила доступності контенту для мобільних пристроїв.....	
3. Модель доступності в Android.....	
3.1	
Дерево та вузли доступності в Android	
3.2 API доступності на Android.....	
3.3 Порядок обходу дерева доступності	
3.4 Семантичні погляди магазину Google Play	
4. Охорона праці	
Висновки	
Перелік посилань.....	
Додаток А	
Додаток Б	
Додаток В	

ВСТУП

Актуальність обраної теми зумовлена унікальною ситуацією, яка склалася сьогодні як в Україні так і у світі. Сьогодні кожен із нас може себе відчутти на місці людини, з обмеженими можливостями. Нас це сердить і нервує, проте ми навіть не задумуємось над тим, що існує величезна кількість людей, які так жили, живуть і будуть жити все своє життя. І їх обмеження не стосуються просто відсутністю можливості поїхати за кордон чи зайвий раз відвідати ресторан. Деякі з них не можуть просто вийти з дому, чи піти на роботу, сходити в магазин, створити сім'ю, тобто зробити ті здавалося б банальні прості речі, які ми в своєму житті часто навіть не сприймаємо за якусь цінність, а звертати увагу починаємо лише тоді, коли втрачаємо цю можливість, а дискомфорт ми відчуваємо навіть при введенні мінімальних тимчасових обмежень.

Відповідно до даних представлених у відкритих інтернет-джерелах, відсоток людей з тимчасовою чи по життєвою інклюзією є достатньо великим. Таким чином, викреслився напрям дослідження. Аналізуючи та розуміючи проблеми цієї величезної групи, обрано об'єкт дослідження, а основні вимоги до функціоналу такого програмного забезпечення встановлено наступні: доступність, інтуїтивна зрозумілість, простота. Варта зауважити, що подібні інструменти в світовій практиці вже існують, проте потрібно розуміти і особливості, менталітет та соціальний статус людей для яких такий продукт розробляється. Адже такі люди доволі часто зустрічають упереджене ставлення, вони оточені стереотипами, їх переслідує дискримінація в житті та побуті, вони весь час знаходяться під загрозою бідності та соціальної не адаптованості і часто проблеми створює не сама людина, а її оточуюче суспільство.

Отже, виходячи із особливостей такого програмного забезпечення, було прийняте рішення скористатися операційною системою Android, яка є ще і платформою для мобільних пристроїв різного типу та ціни, що є надзвичайно

важливим моментом для реалізації поставлених задач, завдяки в першу чергу своїй доступності.

Як і у випадку з іншими платформами, доступність Android дуже високо налаштовується. Значна частина доступності пов'язана з можливістю налаштування, Android має тут величезні переваги. Якщо в Android немає системних налаштувань які відповідають конкретним потребам, то будь-який розробник може створити власну послугу доступності для задоволення цієї потреби. Android дозволяє будь-якому розробнику створити службу доступності. Цю послугу можна використовувати для представлення вмісту на екрані, керування екраном або пристроєм або будь-яке інше завдання, яке ви можете очікувати від допоміжної технології.

Очевидно, що доступність - це не лише те, що приносить користь людям, які мають інвалідність. Вона має набагато ширший діапазон можливостей налаштування для всіх користувачів. Хоча користувачі з обмеженими можливостями можуть отримати найбільше переваг, кожен із користувачів виграє від уваги, яка приділена доступності. У найкращому вигляді доступність - це включення для кожної людини.

Слід подбати про забезпечення порівнянний досвід для всіх користувачів. Дозволити людям для виконання завдань у спосіб, який відповідає їхнім потребам не підриваючи якість контенту. Потрібно надати аудіо описи або роз шифровку відео у додатку. Це зробить доступний оригінальний контент. Мобільні взаємодії мають виняткову якість, дивують і радують своїх користувачів. Слід переконатися, що всі користувачі отримують такий багатий досвід, навіть якщо вони використовують допоміжні технології.

Отже, іноді існують технології, що виключають людей з певними здібностями. Слід поліпшити їх шляхом додавання функції доступності. Але ці доповнення часто відчуються як надбудови вони не є безкоштовні. Ціллю даної роботи є створення технології, щоб кожен відчував себе як вдома. Інвалідність не є бінарним станом. Ми всі маємо здібності і межі цих здібностей. Інвалідність

настає, коли ми створили щось, що не працює для когось з особливими навичками.

1. АНАЛІТИЧНА ЧАСТИНА

1.1. Аналіз доступності

Однією з цілей інклюзивної технології є створення порівнянного досвіду для всіх користувачів. Більшість програмних систем не мають переваги інклюзивного мислення, щоб користуватись винаходом. Як результат середовище не готове сприяти його розробці з доступністю на передньому плані. Усі роки доводилось працювати щоб додати допоміжні технології. Додано багато технологій, таких як телекомунікаційний пристрій для глухих. Служба коротких повідомлень (або SMS), системи розпізнавання голосу та відео дзвінки всі були доступні для телефону. Поки ці допоміжні технології внесли вдосконалення для людей з обмеженими можливостями, проте доповнення до програмних систем не безперервні. З телефоном пов'язана довга історія і він робить пристрій більш зручним для людей з особливими потребами.

Сьогодні телефон як і раніше лідирує з доступними інноваціями. Більшість із нас могли використовувати телефон, як це було призначено, щоб говорити друзям, родиною, підприємствами, у сусідній кімнаті, у сусідній країні чи в іншому місці світу з моменту його винаходу майже 150 років тому. А якщо вашою основною, чи, можливо, єдиною формою спілкування є мова жестів? Це одразу робить телефон непотрібним. Телефон є повсюдно поширеним винаходом, але він представляє вміст в одному носіях – аудіо. Інновації допомогли створити телефон тим із нас, хто не може чути доступніше.

Телекомунікаційний пристрій для глухих часто є альтернативою текстовому вмісту телефону. Клавіатура QUERTY і дисплей принтера, винайдений у 1960-х роках, який підключається до стаціонарного телефону телефон. З допомогою цих доповнень пристрій для глухих дозволяє людям з обмеженими мовою чи слухом,

щоб надрукувати свою розмову. Розмови можуть бути прямий до інших користувачів або до оператора, який передає розмову а не користувач даного пристрою. Хоча телекомунікаційний пристрій для глухих є важливим інструментом для багатьох людей, це не є еквівалент телефону. Якщо ви коли-небудь користувалися одним із цих сервісів, ви обов'язково повинні знати, що взаємодія повільніша та незручніша порівняно зі стандартом телефонного дзвінка, так само, якби ви розмовляли через перекладача на іноземну розмовну мову. Телекомунікаційний пристрій для глухих є доступний, але не включається у функціонал звичайного мобільного пристрою.

FaceTime сьогодні це технологія, яку ми всі вважаємо цілком очевидним. Більшість з нас бере участь при створенні програмного забезпечення і буде використовувати якусь форму системи відео дзвінків майже щодня – Skype, Google Hangouts, Zoom або будь-яка кількість альтернативних систем. Але десять років тому, щоб мати можливість бачити кого завгодно в будь-якій точці світу миттєво у високоякісному відео, і розмовляти з ними було досить дивним. Але для наших глухих колег або людей, які погано чують, це було більш ніж новаторським. Це змінило їхню здатність спілкуватися. Можливість здійснювати розмови, або телефонувати так само. FaceTime став продуктом Apple, тому що Apple задумала створити чудовий інструмент доступності для користувачів жестової мови. Apple вирішила створити чудовий продукт, який буде працювати для всіх. Забезпечивши це працює для всіх і враховуючи доступність протягом усього проекту, Apple супер обслуговує одну конкретну аудиторію.

Мобільний телефон наповнений функціями доступності, такими як FaceTime. Google Помічник і ярлики Siri зменшують вимоги до точності. Це допомагає людям з труднощами в навчанні. Це також зменшує кількість торкання, що вимагають допомоги з моторними проблемами. Екранний час і Safari Reader допоможе звести до мінімуму відволікання – ідеально підходить для людей із розладами дефіциту уваги або ті з проблемами психічного здоров'я, які можуть знайти полегшення у фокусуванні. Диктант, виправлення орфографії, прогнозування тексту, голосові нагадування, вібрація сповіщень, сторонні

клавіатури та підтримка зовнішньої клавіатури всі приклади допоміжних технологій, які ви використовуєте щодня, навіть не розглядаючи їх як функцію доступності. Доступність не повинна бути незрозумілою функцією, яку має лише невелика кількість людей використовувати. У кращому випадку, доступність повинна бути першокласним продуктом або послугою, яку ви створюєте і повинні приносити користь усім. Очевидно, що доступність - це не лише те, що приносить користь людям, які мають інвалідність. Вона має набагато ширший діапазон можливостей налаштування для всіх користувачів. Хоча користувачі з обмеженими можливостями можуть отримати найбільше переваг, кожен із користувачів виграє від уваги, яка приділена доступності. У найкращому вигляді доступність - це включення для кожної людини.

Щоб дійсно зрозуміти, як хтось користується програмою, і які доступності і функціональні можливості потрібні для людей з обмеженими можливостями слід проаналізувати, дослідити і запитати у таких людей. Все це означає – інвалідність не схожа на будь-що в житті, просто бінарний стан. Неможливо розділити світ на дві частини – інваліди та працездатні; або прецедентний приклад, сліпий і зрячий. Є великий розділ між тими з нас, кому доводиться носити окуляри для читання і тими із нас, хто взагалі не відчуває світлового сприйняття, через катаракту, дальтонізм та інші. Порушення зору варіюються від людини до людини з часом, як і всі інваліди.

Отже, інвалідність [1] – це широкий спектр, який ми визначаємо як будь-який орган відсутність або відсутність здатності (внаслідок порушення) до формування діяльності у спосіб або в межах розглянутого діапазону нормально для людини. Іншими словами, інвалідність як особливість цієї особи, щось таке, що відрізняє людину з обмеженими можливостями від усіх нас і нездатні робити те, що ми могли б розумно очікувати від звичайної людини. Проте інвалідність – це не тільки проблема зі здоров'ям[2]. Це складна феномен, що відображає взаємодію між ознаками особистості та особливості суспільства, в якому вона живе. Подолання труднощів, з якими стикаються люди з обмеженими можливостями вимагає утручань для усунення екологічних та соціальних бар'єрів.

Мова йде про системи, на яких ми будуємо суспільства, наприклад, як ми можемо зробити когось із труднощами в навчанні відчувати, якщо ми змушуємо їх завершити велику форму або як хтось із розумовими проблеми зі здоров'ям можуть відчувати себе в пастці приписної системи. Це відомо як соціальна модель інвалідності.

Люди з інвалідністю ідентифікуються меншість, але разом вони становлять одну з найбільших груп меншин у Європі та Північній Америці. У Сполучених Штатах приблизно 27% людей мають інвалідність, це 85,3 млн осіб. Це ставить людей з обмеженими можливостями на один рівень з більшістю населення всіх штати разом. У Сполученому Королівстві повідомляють що є 22% людей з інвалідністю, майже 14 млн. осіб. У всьому світі понад мільярд людей мають інвалідність, приблизно 15% населення планети. Тому є дуже важливим розгляд користувачів системи із різними здібностями. В результаті якість та доступність у програмному продукті, буде набагато вищою. Зрештою, простота і доступність – це правильна річ робити.

1.2. Аналіз області доступності програмних систем у світі

Як і в будь-якому законі, національні варіації законодавства про доступність є величезними. Багато країн взагалі не мають законів про доступність. Там, де існують закони, вони часто стосуються уряду або тільки державного сектору. Іноді це будуть до цифрові закони, грубо відрегульовані за допомогою умов, щоб вони відповідали цифровим каналам. Багато нормативних актів прямо не зосереджені на цифрових областях доступності. Замість цього, вони сконцентровані на більш загальних законах про недискримінацію. Складається ситуація, що людям з інклюзією слід звернутися за консультацією до юриста, щоб визначити, які правила застосовуються до областей в тій чи іншій сфері, які стосуються організації доступності для таких людей.

Сполучені Штати мають один із найстаріших у світі законів про доступність. У 1990 році прийнятий Закон про американців з обмеженими можливостями. Цей закон охоплює вимоги уряду щодо забезпечення доступності

від постачальників і в рамках державних послуг, наприклад, школи. До приватного сектора входять громадські місця – ресторани, театри та магазини для нас із вами. Закон про американців з обмеженими можливостями чітко не говорить про цифровий контент. Але управління юстиції яке підтримує цей закон, стверджує, що він є достатньо широким, щоб керувати цифровий досвід також. У знаковому випадку 2019 року компанія Domino's Pizza вирішила оскаржити це твердження. Подавши справу до Верховного суду, Domino's стверджували, що закон про американців з обмеженими можливостями не стосується їхнього додатку для замовлення піци. Верховний Суд виніс рішення проти Domino's. Це створює прецедент, який означає законні вимоги до мобільної доступності.

Європейський закон про доступність, написаний у 2018 році, є набагато сучаснішим ніж частина законодавства США про американців з обмеженими можливостями. Європейський закон про доступність використовує в якості посилання доступності законодавства з усієї Європи. Він також черпає сигнали з США. Метою є стандартизувати вимоги доступності для цифрової доступності по всій Європі. Цей закон у Європі також прямо не встановлює стандарти доступності для мобільних пристроїв або цифрові послуги в цілому, але він наполягає на мобільній доступності стандартів для визначених категорій. До них належать електронна комерція, банківська справа, програмні послуги, пов'язані з пасажирським транспортом, і програмне забезпечення як частина операційної системи смартфона або самого смартфона.

Вимоги відрізняються залежно від бізнес-категорії додатка. В підсумок, деякі спільні вимоги включають:

1. забезпечення гнучкого збільшення, контраст і колір;
2. забезпечення альтернативи контролю дрібної моторики;
3. надання інформацію в більш ніж одному сенсорному каналі або надання інформації у форматі, який може підтримувати пристрій присутній в альтернативному сенсорному каналі;
4. надання альтернативи для нетекстового вмісту;
5. забезпечення послідовної сумісності із допоміжними засобами технології.

1.3. Демонстрація доступності у програмних системах

Моделювання сліпоти. Операційна система iPhone має функцію доступності для використання разом із вбудованим програмою зчитування з екрана, VoiceOver, що дозволяє користувачам читача зберігати конфіденційність. Викликаний екран Шторка, ця функція вимикає зображення на дисплеї, зберігаючи при цьому сенсорний екран активний. Оскільки користувачі програми зчитування з екрана вже мають свій особистий пристрій читати вголос, це запобігає ненавмисному обміну особистого вмісту візуально. Сліпий або з поганим зором користувач може не знати про приватний вміст навіть якщо він видимий, тому тримати екран увімкненим не має сенсу. Крім того, його вимкнення збільшує конфіденційність і час роботи акумулятора. Екранна чорна завіса дає для нас чудову можливість отримати уявлення про те, що було б за його використання, а так на виході смартфон без візуального зворотного зв'язку. Для цієї діяльності виберіть щоденне завдання, яке ви могли б виконувати за допомогою телефону. Додавання події в календар є хорошим варіантом або зробити її більш релевантною, ви можете вибрати типовий потік з вашого застосування. Увімкнути VoiceOver - прочитайте розділ «Переміщення з допомогою VoiceOver», перш роблячи це. Потім увімкніть екранну завісу, тричі торкнувшись екрана трьома пальцями. Тепер використовуйте програму зчитування з екрана, щоб переміщатися по телефону та виконати завдання, ніколи не відображаючи того, що на екрані. У той час як ті, хто використовує екран щоденно, ймовірно, будуть більш кваліфікованими у його використанні, ніж в середньому інженерна команда, це повинно дати вам певне уявлення про те, які аспекти користування смартфоном важчі, коли голос є єдиним зворотним зв'язком. якщо ви виконуючи це як групову вправу, це також дасть вам уявлення про те, наскільки шумно середовища можуть викликати розчарування при використанні програми зчитування з екрана. Ця програма зчитування з екрана користувача еквівалентно тому, що хтось махає рукою над вашим екраном. Стандартний Android[3] не має еквівалентної функції, але деякі пристрої Samsung

має параметр під назвою Темний екран, який має той самий ефект, що й у режимі зворотного зв'язку.

Моделювання порушення уваги. Для вирішення цього завдання, знадобляться деякі невеликі повітряні кулі або м'які кулі. Вибравши роботу ви робите кожен день на смартфоні стандартний потік з вашого застосування. ваша Задача полягає в тому, щоб завершити цей потік, утримуючи повітряну кулю або кульку в повітрі. У той час як аналогія може бути трохи розтягнутою, це повинно дати вам певне уявлення про те, як може бути легко використовувати ваш додаток для клієнтів із розладами дефіциту уваги.

Моделювання порушень зору. Королівський національний інститут сліпих з Великобританії випустив додаток під назвою eyeWare. Він доступний безкоштовно від компанії Apple App Store або магазину Google Play. EyeWare найкраще працює з Google. EyeWare був розроблений разом з людьми з вадами зору, які описали як це відчувати світ з вадами зору. Додаток дозволяє вибирати з кількох різних видів пошкоджень – від кольору сліпоти до катаракти, глаукоми тощо, включаючи варіації кожного з них. У додатку надається можливість спробувати встановити налаштування та виконайте вибране завдання або завдання на різних рівнях зору.

Отже, іноді існують технології, що виключають людей з певними здібностями. Слід поліпшити їх шляхом додавання функції доступності. Але ці доповнення часто відчуються як надбудови вони не є безкоштовні. Ціллю є створення технології, щоб кожен відчував себе як вдома. Інвалідність не є бінарним станом. Ми всі маємо здібності і межі цих здібностей. Інвалідність настає, коли ми створили щось, що не працює для когось з особливими навичками.

2. ЦИФРОВЕ ВКЛЮЧЕННЯ ІНКЛЮЗИВНОСТІ

Цифрове включення сягає своїм корінням в архітектуру та дизайн продуктів. У процесі створення мобільних додатків необхідний хороший дизайн. Але значну частину доступного досвіду програми все ще створюємо ті з нас, хто не є дизайнерами. Цифрове включення визнає, що кожна людина, в тому числі не обмежуючись дизайнерами, має свою роль у заохоченні більше доступних взаємодій. Навіть якщо ви не дизайнер, я думаю, що варто витратити час на те, щоб висвітлити деякі з основ цієї школи дизайну. Таким чином, можливо зрозуміти деякі з помилок, які були допущені, та заходи, вжиті для їх виправлення. Плюс, якщо ви інженер, ваша робота полягає в тому, щоб втілити проекти в реальність. Отже знання про роботу, яка вплинула на ці проекти, допоможе їх реалізувати.

2.1. Універсальний дизайн

Цифрове включення сягає своїм корінням в універсальний дизайн, рух, який почалося в кінці 1960-х років і привів до багатьох банальних доступності покращення, які ми бачимо навколо себе сьогодні. Архітектор і дизайнер Рональд Мейс придумав фразу універсальний дизайн. Він використав це для опису концепція проектування придатних для використання продуктів і вбудованого середовища, хоча для більшість людей це можливо, в той час як принципово естетично. Архітектор Селвін Голдсміт, який сам перебуває на інвалідному візку,

розвинув концепцію універсального дизайну, відстоюючи сучасний опущений бордюр. Бордюр ний зріз дозволяє інвалідам на візках або людям з проблемами пересування, щоб мати можливість переходити дорогу більш комфортно, ніж зі стандартним піднятим бордюром. Опущений бордюр є також яскравий приклад того, як універсальний дизайн допомагає багатьом людям. Хоча спочатку було призначено для покращення мобільності людей на інвалідних візках, це також дозволяє батькам з колясками та мандрівникам із сумками. Універсальна конструкція являє собою ряд принципів, які використовуються архітекторами і продуктовими та промисловими дизайнерами, щоб направляти їх у досягнення Рональда Мейса. У 1997 році Університет штату Північна Кароліна зібрав деякі з провідні прихильники універсального дизайну, включаючи самого Рональда Мейса.

Разом вони визначили сім принципів універсального дизайну:

1. справедливе використання;
2. гнучкість у використанні;
3. просте та інтуїтивно зрозуміле використання;
4. відчутна інформація;
5. допуск на помилку;
6. низькі фізичні зусилля;
7. розмір і простір для підходу та використання.

Справедливе використання. Дизайн[4] повинен бути корисним і доступним для людей з різними здібностями. Уникайте сегрегації чи стигматизації будь-яких користувачів. Забезпечте однаковий досвід для всіх користувачів, де це можливо. Якщо вам потрібно надати альтернативний досвід, зробіть це еквівалентним. Подбайте про конфіденційність, безпеку та безпеку для всіх своїх клієнтів. Також, переконайтеся, що ваш додаток безперебійно працює з допоміжними технологіями, такими як TalkBack і VoiceOver.

Гнучкість у використанні. Конструкція програмного забезпечення має вмещувати широкий спектр індивідуальних переваг і здібностей. Надайте вибір,

полегшуйте користувачам і адаптуйтеся до них. Приклади цього для мобільних пристроїв включають підтримку різного розміру тексту. Слід дозволити кілька методів введення тексту, наприклад екранну клавіатуру, диктування, або зовнішню клавіатуру. Обидві платформи пропонують ваші різні налаштування клієнта; поважайте їх якомога більше. Приклад гнучкості у використанні наведено на рисунку 1, сторінка 16, а саме надання альтернативи кільком методам введення тексту, з клавіатури та методом диктування.

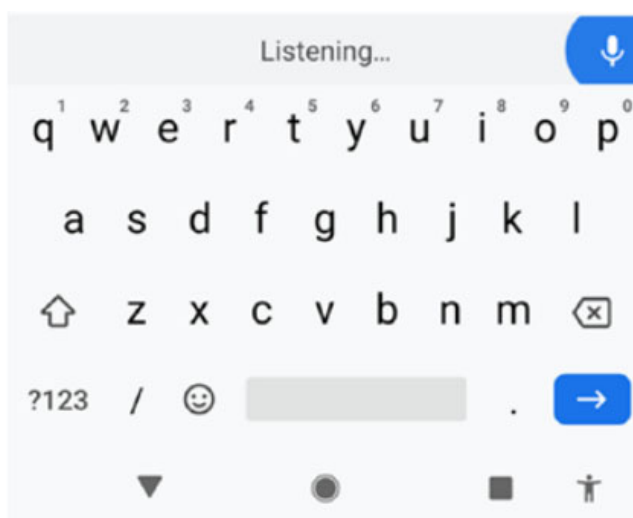


Рисунок 1 – Гнучкість використання у вигляді надання можливості введення тексту через диктування

Просте та інтуїтивно зрозуміле використання. Переконайтеся, що ваш дизайн легко зрозуміти, незалежно від користувацького досвіду, знання, мовних навичок або поточного рівня концентрації. Усуньте непотрібну складність. Будьте відповідні очікуванням користувачів та їх інтуїції. Оволодійте широким спектром грамотності та мовних навичок. Розташуйте інформацію таким чином, щоб вона відповідала структурованості відображення найважливішої інформації на видному місці. Також слід забезпечити ефективні підказки та зворотний зв'язок під час і після виконання завдання. Дотримуйтеся стандартної мови дизайну в усьому додатку, в тому числі дотримуючись наданих системою елементів керування та умов, де це можливо. Ретельно узагальнюйте інформацію, але

дозвольте клієнтам деталізувати її, якщо вони хочуть більше інформації. Зберігайте мову простою і зрозуміло. Коли ваша програма вимагає введення користувача, ви повинні бути чіткими про те, яка інформація вам потрібна і де. Якщо ваш клієнт робить помилку, пояснити це та дозволити їм швидко змінити.

Відчутна інформація. Дизайн повинен ефективно передавати вам необхідну інформацію користувача, незалежно від умов навколишнього середовища або сенсорних здібностей користувача. Використовуйте різні способи (зображення, словесний, тактильний) для надлишкового представлення важливої інформації. Забезпечте достатній контраст між переднім планом і фоном. Забезпечте сумісність з різноманітні методики або пристрої, які використовуються людьми з сенсорними обмеженнями. Щоб відповідати цьому принципу в мобільних пристроях, ми повинні переконатися, що весь зміст є доступний для програми зчитування з екрана пристрою або будь-якої іншої допоміжної технології доступні на пристрої. Уявіть свій зміст в декількох засобах. Використовуйте поєднання кольору, форму, текст зображень і макет разом, щоб зробити ваш зміст змістовним. Приклад забезпечення змістовності наведено на рисунку 2. Програма зчитування з екрана вашого клієнта перетворить текстовий зміст на аудіо зміст, але він не може робити те ж саме з текстом, який ви надаєте як зображення. Щоб отримати цінний зміст, подумайте про те, щоб запропонувати його іншим способом, наприклад як відео.



Accessibility

Screen readers, display, interaction controls

Рисунок 2 – Поєднання кольору, форми, тексту і зображення для забезпечення змістовності

Допуск на помилку. Слід звести до мінімуму небезпеки та несприятливі наслідки випадкових або ненавмисних дій. Розташуйте елементи так, щоб мінімізувати ризики та помилки: Зробіть часто використовувані елементи

керування помітні. Надавайте попередження про небезпеку та помилки. Попередьте свого клієнта заздалегідь, якщо він робить деструктивну дію зміни і дати їм можливість відступити. Якщо ваш клієнт робить помилку, чітко підкресліть це місце і дозвольте їм змінити це без зусиль. Перш ніж робити будь-які дії, дозвольте клієнту переглянути свою відповідь, і дозволити їм змінити або повернути. Забезпечте такі руйнівні засоби безпеки, щоб не було можливості виникнути випадково. У мобільних додатках це включає використання чіткої та короткої інформації про наслідки кожної дії, коли ви просите свого користувача прийняти рішення. Додайте вібрацію до шкідливих або руйнівних дій і надайте зворотній зв'язок своїм клієнта про те, чого очікувати, коли вони приймуть рішення.

Низькі фізичні зусилля. Переконайтеся, що клієнти можуть використовувати ваш дизайн ефективно, зручно та отримавши мінімум втоми. Зведіть до мінімуму повторювані дії та постійні зусилля. Усі кнопки на вашому інтерфейсі повинні бути легкодоступними, особливо на пристроях з великим екраном(рисунок 3). Надайте ярлики до місць загального користування вашого додатку і підтримуйте Google Assistant або ярлики Siri, де це можливо.

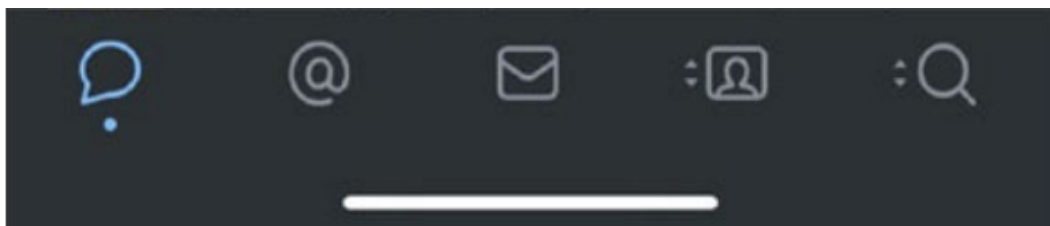


Рисунок 3 – Надання легко доступності функціоналу

Розмір і простір для підходу та використання. Остаточний принцип переважно підходить для фізичних предметів. Принцип стверджує, що слід забезпечити належний розмір і простір для підходу, охоплення, маніпулювання і використання незалежно від розміру тіла, позицій, або мобільності користувача. Зробіть доступ до всіх компонентів зручним і врахуйте варіації в руці та розмірі. Виконавши цю інструкцію, є можливість зробивши інтерфейси адаптованими до

різні розміри екрану та розміри тексту. Переконайтеся, що ваш додаток працює з Switch керуванням та комутаційним доступом, а також можливістю навігації з клавіатури; Також з голосовим керуванням та голосовим доступом, дозволяючи своїм клієнтам керувати своїм додатком практично без жодних рухів. Прикладом цього є рисунок 4, сторінка 19, процес забезпечення керування на Android за допомогою голосового доступу. Будь-які інтерактивні квадрати елементів повинні бути не менше 44px, щоб не налягати на дрібній моториці контролю.

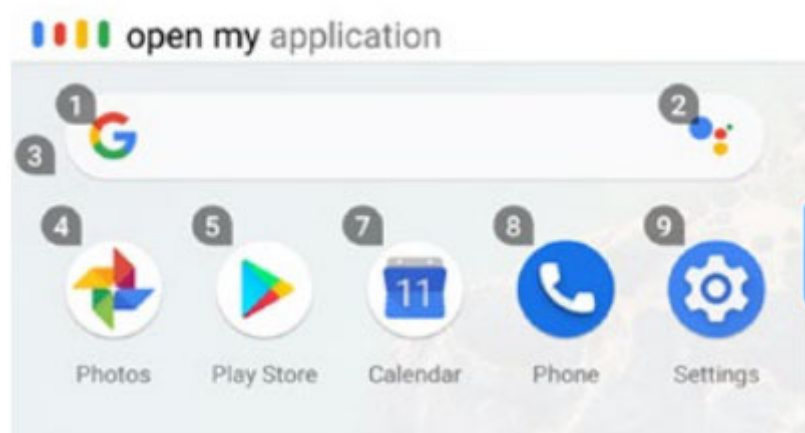


Рисунок 4 – Функція голосового доступу на Android

Ці універсальні принципи проектування призведуть до створення багатьох деталей на які звертається увага до доступності кожен день. Розглянемо аудіокниги, автоматичні двері, висококонтрастні вивіски та гнучкі трубочки для пиття. Всі ці продукти були створені за такими ж універсальними принципами проектування. Вони служать багатьом людям з різними здібностями, але є поліпшення для всіх також інше. Це те, до чого слід прагнути в програмах. Існуюча ідеальна мета – це створити досвід, який охоплює всіх.

Універсальний дизайн враховує інвалідність[5] і враховує те, що ви можете досягти зміни дизайну, щоб допомогти людям із цією інвалідністю, тим самим знижуючи бар'єри для використання продукту. Зниження бар'єрів розширює ринок для потенційних користувачів. Втім, як універсальний дизайн був розроблений в епоху до цифрових технологій фізичні продукти.

2.2. Інклюзивний дизайн

Інклюзивний дизайн поширюється на переваги універсального дизайну. Це є як правило, більш пристосований до цифрових взаємодій. Поки універсальний дизайн розглядає інвалідність, інклюзивний дизайн враховує індивідів і тенденцій до більш доступного першого підходу. Інклюзивний дизайн робить великий крок вперед у визнанні, що інвалідність не є постійним, двійковим станом. Нерівність у здібностях людей типова і різниться як функція часу чи ситуації. Інклюзивний дизайн робить це шляхом врахування потреби людей, як постійні, тимчасові, ситуаційні, або зміни. Покращуючи досвід для людей із особливими потребами та враховуючи ці потреби в першу чергу, ми можемо розширити цю перевагу працювати для ширшого населення, таким чином покращуючи досвід для кожного у певний момент свого життя. Консультант з цифрової доступності The Paciello Group допоміг визначити сім принципів інклюзивного дизайну:

1. надати порівняльний досвід;
2. розглянути ситуацію;
3. бути послідовними;
4. дати контроль;
5. пропонувати вибір;
6. розставити пріоритети вмісту;
7. додати цінність;

Надати порівняльний досвід. Переконайтеся, що ваш інтерфейс забезпечує порівняльний досвід для всіх. Слід дозволити людям для виконання завдань у спосіб, який відповідає їхнім потребам підкреслюючи якість контенту. Ви можете надати аудіо описи або розшифровку відео у вашому додатку. Це зробить ваш контент доступним і оригінальним. Але чи ця альтернатива вловлює суть і тон оригіналу? Мобільні взаємодії мають виняткову якість дивують і радують своїх користувачів. Ви повинні переконатися, що всі ваші користувачі отримують такий багатий досвід, навіть якщо вони використовують допоміжні технології.

Прикладом є регіони в реальному часі Android. Живі регіони – це область екрана, яка автоматично повідомляє допоміжні технології, коли вміст змінюється. TalkBack читатиме цей вміст без необхідності переходити на нього цей вміст. iPhone OS не має функції живого регіону, але ви можете створити її. Ви також можете використовувати сповіщення про спеціальні можливості.

Розглянути ситуацію. Люди використовують свій інтерфейс в різних ситуаціях. Переконайтеся, що ваш інтерфейс надає цінний досвід людям, незалежно від їхніх обставин. Подумайте про використання свого телефону при яскравому сонячному світлі, аудіо та тактильних відчуттях зворотний зв'язок разом із високо контрастними кольорами, які допоможуть у цій ситуації. Надання субтитрів або закритих субтитрів у вашому відео вмісті дозволяє батькам дивитися відео зі зниженим звуком, не заважаючи їхніх сплячих дітей.

Бути послідовними. Використовуйте знайомі конвенції та застосовуйте їх послідовно для просування знайомства і розуміння. Ця послідовність застосовується як у вашому додатку і проти системи, на якій ви працюєте. Не винаходьте велосипед у своєму дизайні. Дизайн встає на плечі гігантів за допомогою елементів управління, Android і iOS надає вам їх, не соромтеся створювати підкласи елементів керування, щоб додати функцію або зовнішній вигляд, який вам потрібен, але починаючи з них як основи, ви отримаєте це багато додаткового безкоштовно. Ваш користувач буде відчувати себе як вдома завдяки його взаємодії відчує себе знайомим, а Apple і Google вже багато чого написали з функцій доступності.

Дати контроль. Переконайтеся, що люди контролюють. Ваші клієнти повинні мати доступ і взаємодіяти з вмістом у свій улюблений спосіб. Не придушуйте або вимкните можливість змінювати стандартні налаштування платформи, наприклад орієнтацію або розмір шрифту. Крім того, уникайте змін вмісту, яких немає ініційовано вашим користувачем, якщо немає способу керувати ним. Як розробник є способи відключити деякі допоміжні функції. Наприклад, ви можете створити програму, яка не підтримує масштабування розмірів тексту. Це може означати, що ви отримаєте ідеальний піксельний дизайн,

але це значно вплине можливість вашого користувача зручно використовувати ваш додаток.

Пропонувати вибір. Подумайте про те, щоб люди могли виконувати завдання різними способами, особливо завдання, які є складними або нестандартними. Часто існує більше одного способу виконати завдання. Ви не можете припустити, якому методу хтось може віддати перевагу. Надаючи альтернативи для макета та виконання завдання, ви запропонуєте людям вибір, який підходить їм і їхнім обставинам на даний момент. Чудовим прикладом цього є видалення електронної пошти в програмі «Пошта» для iOS. Я можу це зробити кнопкою, проведенням пальцем або довгим натисканням. Це дозволяє мені, як користувачу, вибрати варіант, який є найшвидшим для того, як я використовую програму.

Розставити пріоритети вмісту. Допоможіть користувачам зосередитися на основних завданнях, функціях та інформації, розставляючи пріоритети їх у змісті та макеті. Визначте основне призначення інтерфейсу, а потім вміст і функції, необхідні для досягнення цієї мети. Інтерфейси можуть бути складними для розуміння, коли основні функції відсутні чітко висвітлені та визначені пріоритетами. Логічно структуруйте свій вміст і позначайте заголовки ознакою заголовка або ролі. Коли ви запускаєте програму електронної пошти, вам не відображається список поштові скриньки та папки на вибір. Вам відображається папка "Вхідні", тому що це і так найістотніша ознака. Стандартні функції повинні бути простими і завжди доступними. Більш складні дії повинні бути можливі, але не видатними. Якщо зробити все доступне відразу, інтерфейс буде заплутаним. Розгляньте можливість розміщення розширених елементів керування за допоміжним жестом, як-от довге натискання.

Додати цінність. Розглянемо цінність функцій і те, як вони покращують роботу для різних користувачів. Зверніть увагу на такі функції пристрою, як голос, геолокація, камери та API вібрації та як інтеграція з підключеними пристроями або другий екран може забезпечити вибір. Це область інклюзивного дизайну, де мобільний стоїть на голові і плечі вище інших платформ. Мобільні

пристрої переповнені датчиками і режимами зворотного зв'язку. Зовнішні пристрої, такі як розумний дім, екрани або зовнішні контролери можна підключати бездротовим способом. Елементи керування на обох платформах використовуйте тактильні засоби, щоб забезпечити тактильний зворотний зв'язок користувачам, коли щось має змінено. Приклад цього відображено на рисунку 5, сторінка 23 де зображається тактильний зворотній зв'язок. Хоча це також приємний досвід, це є необхідний режим зворотного зв'язку для клієнтів з вадами зору, допомогу вони визначають, коли вони внесли зміни.

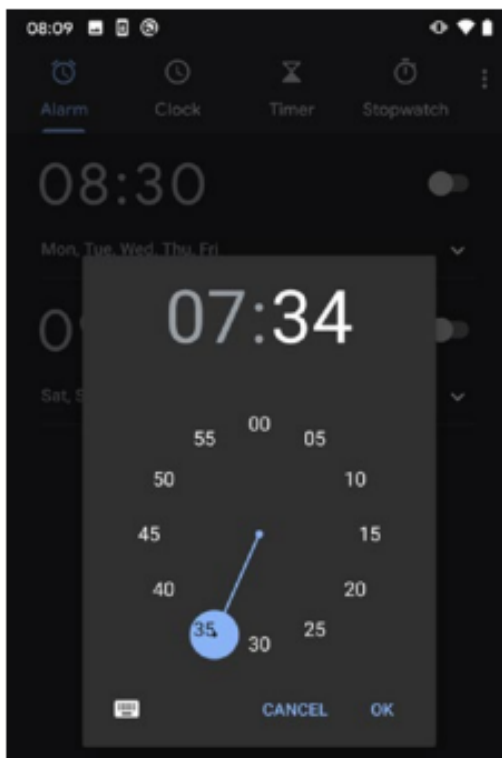


Рисунок 5 - Забезпечення тактильного зворотного зв'язку при зміні хвилин та
ГОДИН

2.3. Цифрове включення

Важливою частиною цифрової інтеграції є усвідомлення того, що люди з обмеженими можливостями, хоча і в основному, але це не єдина меншина, яка зможе скористатися сервісом. Інші фактори, такі як вік, стать, освіта, соціальне

життя угруповання та доходи також можуть вплинути на те, які будуть необхідні технології для розробки. Підсумуємо принципи, які складають цифрову інклюзію:

1. емпатичний;
2. ситуаційний;
3. інституційний.

Емпатичний принцип. Зважаючи на інших і усвідомлюючи, що кожен є особистістю з різним рівнем знання, досвідом і здібностями, завдяки цьому люди сприймають вашу програму по-різному – можливо, це найважливіший принцип цифрового включення.

Ситуаційний принцип. Не всі перешкоди для використання програми чи сервісу виникають через фізичні обставини здібності. Інклюзія означає розгляд ситуації людей в цілому. Це включає їхні навички, а також інші фактори: фінансову, гендерну, цифрову грамотність, психічне здоров'я та першу мову. Отже, всі ці фактори впливають на те, як хтось може використовувати додаток.

Інституційний принцип. Програмне забезпечення, яке створює організація, відображає саму організацію. Розглянемо, як майже лібертаріанська точка зору деяких лідерів соціальних мереж привели до платформ, на яких може бути майже все, в тому числі іноді расизм, втручання у вибори та переслідування. Усе в організації може зіграти роль у покращенні доступності програми для користувачів програми і, можливо, несе відповідальність за це.

2.4. Аналіз профілів людей для визначення цифрового включення

Чудова техніка, яка допомагає розглядати користувачів як осіб створюють профілі. Розглянемо те, що людина хоче зробити з додатком і чому. Думки індивідуальної людини обставини, досвіду і здібності є кардинально особисті. Найголовніше, присвоїти конкретній людині ім'я та називати її цим ім'ям як майбутнього користувача системи для створення цифрових включень. Використаємо таблицю 1, щоб визначитись з потребами функціоналу майбутнього користувача.

Таблиця 1 – Створення картини користувача з інклюзією

Назва пункту	Характеристика
Ім'я особи	Важливо надати вибір ім'я для створення картини
Вік	Для визначення приблизного вікового діапазону
Стать	Слід пам'ятати, що близько 2% населення не ідентифікуються за статтю.
Найближчі родичі	Чи є утриманці, діти чи дорослі, які про них піклуються? Чи одружені, живуть з сім'єю, або, можливо, живуть одні?

Продовження таблиці 1

Робота і дохід	Чи безробітні, працевлаштовані, працюють з неповною зайнятістю або надзвичайно успішні у своїй діяльності чи будують кар'єру? Можливо, ви переможець лотереї? Слід розуміти стан для визначення потрібного функціоналу.
Інвалідність	Близько 14% людей у всьому світі мають інвалідність. Слід прагнути до різних профілів, які відображають різні інвалідності або ж взагалі жодної.
Передісторія	Слід пам'ятати, що профіль призначений для представлення справжньої людини, яка повинна розповісти свою історію, а не соромитися, це не допоможе визначити функціонал для допомоги людину з інклюзією.
Для чого цей додаток?	Що для цієї людини особисто зробить додаток? Це їхній вибір, чи вони з якоїсь причини будуть використовувати додаток?

Люди з обмеженими можливостями повинні мати еквівалентний досвід з іншими. Вони хочуть мати можливість використовувати свою програму так само, як і будь-хто інший. Доступність це не жест для того, щоб почуватися краще або

щоб завоювати вплив бренду. Доступність - це інструмент для включення у ваше програмне забезпечення.

Слід розглянути практичну сторону, як ми можемо допомогти різним людям, які хочуть використовувати програми. Охоплюючи будь-які конкретні технології, слід досягнути цілей. Правила доступності веб-контенту, є остаточною структурою для того щоб визначитись, які покращення слід внести в програми для того, щоб вони стали більш доступні.

2.5 Інструкції доступності контенту для мобільних пристроїв

Існує багато принципів які мають вказівки, за якими слідують критерії успіху і техніки. Спочатку рекомендації були написані спеціально для інтернет контенту. Але більшість принципів і рекомендацій є загальними і їх слід дотримуватись при мобільній розробці. Стандарт ISO з 2012 року, є основним з багатьох законів про доступність у всьому світі. Він часто використовується як еталон щоб суди вирішували, чи доступний цифровий досвід у тому чи іншому додатку чи ні, заради позовів про доступність. Таким чином, що стосується закону, то правила доступності для веб-контенту також стосуються контенту мобільних пристроїв.

2.6 Правила доступності контенту для мобільних пристроїв

Сформуємо загальні поради щодо того, як зробити так, щоб відповідати кожній інструкції щодо мобільних пристроїв і тим, які інструменти слід використовувати для дотримання правил доступності контенту.

Першим і дуже вагомим правилом є помітність. Інформація та компоненти інтерфейсу користувача повинні бути присутніми користувачам так, як вони можуть це сприймати. Перший принцип, зрозумілий, охоплює зміст вашої заявки та способи, якими ви робите це доступним для вашого клієнта. Узагальнення за цим принципом одним словом «альтернативи». У тому ж ключі, якщо ви надаєте вміст лише візуальний формат, наприклад зображення, текст, недоступний на

екрані або відео з субтитрами без аудіо доріжки, з якими ваші клієнти люди з вадами зору не матимуть доступу до цього вмісту.

Другим правилом є текстові альтернативи. Надайте текстові альтернативи для будь-якого нетекстового вмісту, щоб він міг змінюватися на інші форми, потрібні людям, наприклад, великі шрифти, шрифт Брайля, мова, символи або простіша мова. Переконайтеся, що весь нетекстовий вміст має відповідну і еквівалентну текстову альтернативу. Часто це відноситься до значних зображень. Надання опису до декоративних зображень може надати вам більш багатий досвід користувачів. Але будьте обережні, щоб не відволікати від вмісту вашої програми.

Можливість додати текстову альтернативу майже до будь-якого об'єкта забезпечена на обох платформах мобільних додатків як на Android так і на iOS. Android використовує властивість `contentDescription` зображені на лістингу 1 та лістингу 2.

Лістинг 1 - Налаштування Android `contentDescription` у файлі XML макета

```
<Button  
...  
android:contentDescription="Submit" />
```

Лістинг 2 - Встановлення опису вмісту Android у коді

```
submitButton.contentDescription = "Submit"
```

Альтернативний текст для зображень. Зображення, які мають функцію інтерфейсу користувача, наприклад кнопку, повинні мати чіткі, альтернативні мітки описового тексту, щоб користувачі програми з екрана могли знати про її наявність, доступні дії та які будуть наслідки цієї дії. Декоративні зображення

вимагають більшої уваги. Невеликі декоративні зображення наприклад значки, що привертають увагу до вмісту в рядку таблиці, ймовірно, не повинні містити альтернативний текст. Це додає непотрібний шум, збільшує кількість пальців, необхідних для навігації, і заплутує ваш порядок інтерфейсу, що призводить до втрати контексту та уваги вашого користувача. Більші декоративні зображення, такі як зображення героїв, однак, ймовірно, повинні містити текстовий опис призначення зображення – наприклад, піктограма галочки має містити опис «все готово», а не «значок галочки». Ця мітка ніколи не повинна відводити фокус від верхнього колонтитула сторінки, а натомість служити ознакою для користувачів з слабким зором, які намагаються щось знайти на екрані. Якщо програма зчитування з екрана пропускає велику область на екрані, ці користувачі можуть вважати, що ця частина екрана містить не дороблений функціонал і намагатимуться знайти в ньому будь-який зміст та смисл. Це призведе до дратівливого стану, якщо ви не опишете що присутній відступ у контенті. Думайте про це як про еквівалент програми зчитування з екрана «ця сторінка залишена навмисно порожньою».

Третім правилом доступності контенту для мобільних пристроїв є надання альтернативи для медіа, ЗМІ, що засновані на часі. Для нас із вами це означає аудіо та відео. Крім того, він охоплює будь-які медіа, що має часовий компонент, наприклад анімацію. Відео та аудіо має містити підписи або транскрипцію. Бажано включати відео, аудіоопис, а в ідеалі як аудіо, так і відео мають включати можливість мови жестів. Ця інструкція виключає відео або аудіо, які ви надаєте як альтернативу до тексту. Якщо ви надаєте їх як альтернативу, обов'язково позначте їх як такий. Якщо ви не даєте зрозуміти користувачам, що ваші медіа є альтернатива, люди, які не можуть використовувати це, можуть відчувати себе виключеними.

Четверте правило доступності контенту мобільних пристроїв - створюйте контент, який можна представити різними способами (для приклад простішого макета) без втрати інформації або структура. Адаптовані вказівки відходять від змісту й охоплюють, як представити вміст у інтерфейсі користувача. Подавайте

вміст у чіткому, змістовному порядку. Ви можете цього досягти завдяки гарному дизайну інтерфейсу, але подбайте про тестування за допомогою програм зчитування з екрана. Вони можуть часто переміщатися по екрану таким чином, щоб не могло бути природні для зрячих користувачів. Наприклад, розглянемо екран із великим героєм зображення з назвою. Ваше око зазвичай проглядає зображення і стає притягнуто до заголовка. Програма зчитування з екрана зосередиться на альтернативному тексті, наданому вашому зображенню спочатку, якщо ви не вкажете, що заголовок має бути спочатку сфокусованим. Подання інформації в різних сенсорних характеристиках. Не варто покладатися на один колір, наприклад, але комбінація таких атрибутів, як колір, форму, розмір, розташування або звук. Ви можете використовувати будь-який або всі з цих режимів разом, щоб зробити висновок. Поля введення та інші елементи керування повинні визначати мету або дію для цього контролю. Використовуйте властивість `contentDescription` на Android та властивість `accessibilityLabel` на iOS, про які ми раніше говорили досягти цього. В iOS функції доступності є потужним інструментом для інформування допоміжної технології та її користувачів про те, як буде функціонувати елемент керування. Підказка для Android (лістинг 3) може надати додатковий контекст, якщо потрібно.

Лістинг 3 - Встановлення підказки в XML-файлі макета Android

```
<EditText  
...  
android:hint="Email Address" />
```

Використовуючи описи, мітки чи підказки для надання контексту, може бути спокусливим щоб надати занадто багато фону, наприклад, додавши «двічі торкніться, щоб редагувати» як підказка до текстового поля. Пам'ятайте: користувачі програми зчитування з екрана будуть так само природно використовувати екран інтерфейс читача, як ви чи я, використовуючи візуальний

інтерфейс. Поки ці відмінності в навігації можуть бути для вас новими, пояснювати їх не потрібно. це додає непотрібного шуму і може здатися покровительським для ваших користувачів.

П'яте правило - зробіть користувачам легшим перегляд і чуття вмісту відокремлюючи передній план від заднього. Ця інструкція містить основні правила для контрасту, розміру тексту та міжрядкового інтервалу, фонового звуку тощо. Деякі недоліки, такі як дальтонізм, можуть перешкоджати нашій здатності розрізняти елементи. Тому дотримуйтеся цих посібників і дозволяйте настроювання, де це необхідно, це є важливими. Якщо цього не вимагає ваш макет уникайте прокручування в двох вимірах, оскільки це може стати тригером для людей чутливі до руху. Це може бути складно контролювати за допомогою деяких допоміжних засобів увімкнені технології.

Використання кольору в будь-якому додатку має вирішальне значення. Правильний вибір палітри може бути важливим фактором, що виділяє ваш додаток. Добре використаний колір може бути як тонкий натяк на значення і допомагає зрозуміти що це за елемент. Однак пам'ятайте про що не у всіх відчуття кольору однакові. Тому колір повинен ніколи не бути єдиним способом ідентифікувати елементи або передати значення. Комбінуйте розфарбовування текстом, фігурами або анімацією. Жовті попереджувальні трикутники, для прикладу, вони називаються не просто «жовтими попередженнями», а частина трикутника важливо для його корисності. Коефіцієнт контрасту між кольором вашого тексту та фоном тексту колір — це матеріальний фактор. Великий текст – визначається як мінімум 18pt або 14pt напівжирний – повинен мати мінімальний коефіцієнт контрастності 3:1. Менший текст повинен мати коефіцієнт контрастності 4,5:1. В ідеалі ваш текст повинен мати а коефіцієнт контрастності 7:1, з більшим текстом 4,5:1. Деякі люди, наприклад ті, хто страждає на дислексію, синдром Ірлена, або дальтонізм, може скористатися можливістю налаштування фону кольорів. Це було б чудовим доповненням до вашої програми, якщо ви хочете взяти ваша доступність на наступний рівень.

Будь-який звук у вашому додатку, включаючи аудіо як частину відео, тривалістю більше ніж 3 секунди повинен мати контроль. Вони повинні дозволяти зробити паузу, зупинку, і регулювання гучності. Регулювання гучності має бути незалежним від контроль системи, щоб дозволити вашому клієнту підтримувати вибраний системний рівень. Будь-яке звукове мовлення має уникати фонових звуків. Крім того, дозволити вимкнути фонові звуки або мати фонові звуки щонайменше на 20 дБ (принаймні в чотири рази) тихіше, ніж звук мовлення. Фоновий шум може вплинути на людей з розладами уваги; це також може зробити аудіо переднього плану складним для визначення для деяких людей з порушення слуху, якщо їх слух спотворений.

Розмір тексту має бути змінений до подвоєного розміру гліфа без втрати вмісту або функціональність. Забезпечення підтримки системи, динамічних розмірів тексту на обох платформи відповідатимуть цій вимозі. Уникайте використання зображень тексту як таких не підтримують програми зчитування з екрана або налаштування розміру тексту. Якщо ви повинні використовувати зображення, що містить текст, переконайтеся, що ви додали мітку, доступну для зчитування з екрана включаючи текст до зображення. Зробіть це за допомогою `contentDescription` властивість на Android або `accessibilityLabel` на iOS. Під час форматування тексту дотримуйтесь наступного: уникайте повного вирівнювання, не більше 80 символів у рядках тексту та надайте абзац відстань у 1,5 рази більше міжрядкового інтервалу. Дозвольте вашим користувачам налаштувати їх потреб, забезпечуючи механізм зміни кольорів тексту та фону.

Шосте правило доступності контенту для мобільних пристроїв - зробіть всі функції доступними з клавіатури. Переконайтеся, що ваш додаток доступний для навігації за допомогою зовнішньої клавіатури. Переконайтеся у вашій програмі немає клавіатурних пасток – елемент керування, на який можна сфокусуватися клавіатури, але втратить фокус лише іншим способом. Еквівалент може також трапляється під час навігації по інтерфейсу за допомогою програм зчитування з екрана. Бути впевненим щоб перейти до кожного елемента керування за допомогою програм зчитування з екрана та зовнішньої клавіатури під час

тестування вашої програми. Обидві платформи підтримують клавіатурну навігацію, але реалізація кожної відбувається по-різному.

Сьоме правило - надайте користувачам достатньо часу для читання та використання вмісту. Часові обмеження можуть бути бажаними з багатьох причин: щоб безкоштовні продукти або ресурси, що запобігають атаці відмови в обслуговуванні або додають безпеку для вашого користувачів, обмежуючи несанкціоноване використання маркерів авторизації. Однак, часові обмеження можуть створити перешкоду для деяких людей. Навігація по програмі за допомогою допоміжної технології може зайняти багато часу, особливо якщо вона доступна дизайну не по стандарту. Крім того, люди з тривожними розладами можуть вважати часові обмеження дуже напруженими. Якщо можливо, уникайте обмежень у часі, якщо немає суворих вимог щодо цього. Якщо ви все-таки включаєте обмеження часу, подумайте про надання механізму для користувач може вибрати обмеження або розширити його. Попереджайте, перш ніж настане ліміт щоб завершити термін дії, і дайте можливість скинути або продовжити. Якщо повторна аутентифікація є то дзвольте клієнту продовжити з того місця, де він зупинився термін аутентифікації. Надайте механізм для паузи, зупинки або приховати їх на вимогу користувача.

Восьме правило - не створюйте вміст таким чином, що, як відомо, може викликати судомі або фізичні реакції. Будь-яке блимання або мигання не повинно блимати більше трьох разів на раз. Відомо, що це викликає фізичні реакції у людей з епілепсією. Крім того, ви повинні надати клієнту можливість вимкнення анімації, якщо це не важливо для функціональності вашої програми. На iOS, вам слід прослухати властивість `isReduceMotionEnabled`.

Дев'яте правило - зробіть текстовий вміст читабельним і зрозумілим. Уникайте ідіоми, жаргон і скорочення в тексті, де це можливо. В будь-якому випадку використовуйте, механізм для вашого користувача, щоб визначити їх визначення. В ідеалі, намагайтеся, щоб будь-який письмовий вміст був зрозумілим і нижчий рівня середньої освіти.

Десяте правило - появлення усіх компонентів додатку повинне працювати у передбачуваний спосіб. Будь-які зміни контексту – переміщення чи зміна значення чи поведінки елементів керування або вмісту, наприклад переходу на новий екран – не повинно відбуватися без попередження. Вони мають відбуватися лише на прохання користувача, скажімо, натиснувши кнопку. Або ви повинні повідомити про це свого клієнта що щось трапиться, можливо, із завантажувальною флешкою. Відмова підготуватися користувач через зміну контексту може викликати занепокоєння у деяких клієнтів з психічними проблемами або труднощами в навчанні. Крім того, візуально користувачі з порушеннями можуть не знати, що контекст змінився, таким чином сенс нового змісту стає незрозумілим.

Одинадцяте правило - допоможіть користувачам уникнути та виправити помилки. При зборі даних від клієнта через форму, будь-які помилки повинні бути чітко виділені на місці. Надайте вказівку на те, що є причини помилки та як це можна виправити. Кожне поле має забезпечити В ідеалі будь-які форми повинні забезпечувати принаймні одну з наступних характеристик:

1. Реверсивність – є можливість скасувати подачу даних;
2. Перевірка – введені дані перевіряються перед подачею, користувач має можливість виправити будь-які помилки;
3. Підтвердження – надайте користувачеві можливість переглянути, підтвердити та виправити будь-які введені дані перед завершенням їх подання.

Дванадцяте правило – вміст має бути достатньо надійним, щоб його можна було інтерпретувати. Остаточний принцип містить лише одну настанову. Основне правило з яким програма має бути сумісною з будь-якою доступною технологією вибраною користувачем платформи.

Останнє правило доступності контенту для мобільних пристроїв - максимальна сумісність із поточними та майбутніми засобами та технологіями. Мета ваших елементів керування повинна бути визначена вашим користувачем. Ви можете досягти цього, використовуючи риси доступності на iOS. Наприклад,

будь-який елемент керування, який діє як кнопку, має бути позначений в прикладі. Це дозволяє допоміжні технології у використанні, щоб прийняти рішення про те, як вести себе для цього елемента.

Отже, Інструкції щодо доступності контенту мобільних пристроїв встановлюють правила щодо того, як має вимірюватись доступність вашого додатка. Ці стандарти і формують підґрунтя для визначення, чи доступна програма. Якщо ви не відповідаєте критеріям доступності, виникає ризик до юридичних спорів. додаток має бути зрозумілим, працездатним, зрозумілим і міцним.

3. МОДЕЛЬ ДОСТУПНОСТІ В ANDROID

3.1 Дерево та вузли доступності в Android

Перш ніж розглянемо функції доступності, доступні на Android, давайте розглянемо як працює модель доступності Android для допоміжних технологій. Android створює таке відоме явище, як дерево доступності, ієрархічне представлення елементів, які присутні на екрані. Допоміжні служби можуть використовувати це дерево доступності для визначення способу презентації інформації. Android робить деякі розумні припущення щодо вашого користувальницького інтерфейсу та як представити його в допоміжній технології. Більшість з часу, це дасть доступний досвід функціонування. Іноді потрібно буде налаштувати це дерево, щоб представити кращий досвід.

Android надає дані службам доступності у вигляді дерева вузлів доступності. Вузли доступності є представленнями елементів на екрані. Вони містять вміст і метадані про елемент і дії, які можуть виконувати допоміжні технології. Різниця між вузлом доступності та елементом екрану має вирішальне значення. Android не має допоміжних технологій елементів на екрані, але проксі-об'єкт, що містить інформацію про актуальні об'єкти для доступності. Після того, як служба доступності отримує вузол, цей вузол стає незмінним. Зміни у вигляді не розкриваються доступність до наступного разу, коли служба запитає вузол. Вузол доступності містить інформацію про елемент, який є корисним для допоміжних технологій, наприклад, таких властивостей, як текст і опис, інформація, наприклад, чи елемент можна прокручувати або чи включений заголовок. Так є відомості про стан елемента на екрані, це і є ієрархія.

Android дозволяє будь-якому розробнику створити службу доступності. Отримання дерева доступності зображено на рисунку 6, сторінка 35. Цю послугу можна використовувати для представлення вмісту на екрані, керування екраном або пристроєм або будь-яке інше завдання, яке ви можете очікувати від

допоміжної технології. Android дійсно забезпечує усі тонкощі створення такого сервісу. Давайте розглянемо усі основи, як працює дерево доступності .

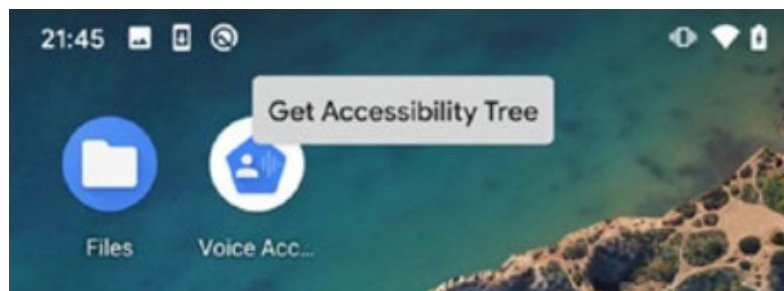


Рисунок 6 – Отримання дерева доступності на Android

3.2 API доступності на Android

API доступності Android відповідають за керування представленнями службам доступності. Google зробив деякі розумні припущення при створенні своїх стандартних елементів керування. Дотримуйтесь Android класів для елементів керування, і ваша доступність буде в порядку. Але іноді потрібно буде внести зміни, щоб покращити те, що ми представляємо користувачам як доступність. Це особливо актуально, коли ви створюєте свій власний спеціальний елемент керування. Якщо ви змінюєте вбудовані елементи керування Android, це важливо для вас, щоб ви не видалили жодних функцій доступності. Розглянемо деякі властивості, які можуть знадобитися для дотримання доступності. Важливим для доступності є те, що якщо елемент є суто декоративним, наприклад, піктограма, що супроводжує текст, додавання цього елемента у ваше дерево доступності означає додаткові непотрібні свайпи для користувачі TalkBack. У цьому випадку ми не повинні надавати цей елемент допоміжному технології. Ви можете зробити це, встановивши для параметр `importantForAccessibility` із значенням `no` (лістинг 4 та лістинг 5). Крім того, якщо елемент, який зазвичай `hidden`, ви можете встановити значення `yes`.

Лістинг 4 – Налаштування перегляду недоступного в XML

```
<TextView
```

```
...
```

```
android:importantForAccessibility="no" />
```

Лістинг 5 – Налаштування перегляду недоступного в коді

```
textIcon.importantForAccessibility = IMPORTANT_FOR_ACCESSIBILITY_NO
```

Значення підказки – це текстове представлення елемента, окремого від елемента текстового значення (Лістинг 6). Ці значення по-різному використовуються Android, залежно від того, має ваш елемент інше текстове значення чи ні. Для елемента без текстового значення підказка відображається як значення заповнювача. Підказка також зчитується TalkBack як текстова мітка. Це корисно для елементів EditText, щоб надати користувачам підказку для якої інформації вони повинні вводити. Для цих цілей ваша підказка має бути короткою і змістовною, дотримуючись тих самих черг, що й для contentDescription вище. Для елементів із текстовим значенням підказка не відображається. TalkBack читає підказку після текстового значення елементів або contentDescription і типу element. Ви можете використовувати підказку, щоб надати користувачеві більше контексту, якщо це допомагає вони переміщуються по вашому інтерфейсу. Ідеальним варіантом використання було б дати короткий опис наслідком натискання кнопки, наприклад. Кнопка з текстом "Надіслати" може мати натяк на "Надсилання вашого повідомлення". Це робить зрозумілим, яка дія буде виконана і який очікуваний результат. Як посібник для написання підказок, уявіть, що пояснюєте другові, що це за елемент керування робить.

Лістинг 6 – Встановлення підказки в XML-файлі макета

```
<EditText  
...  
android:hint="Email Address" />
```

Встановлення заголовка доступності. Під час навігації по екрану за допомогою TalkBack користувачам, у яких немає або дуже мало зорової навігації, проводячи зліва направо по екрану важко концентруватись. Для екранів із великою кількістю вмісту це може означати велику кількість гортань через невідповідний вміст, щоб клієнт міг знайти потрібну інформацію шукаю. Позначення ваших заголовків як на лістингу 7 та лістингу 8 дозволяє користувачам TalkBack переглядати заголовки, пропускаючи вміст.

Лістинг 7 - Встановлення елемента як заголовка у XML-файлі макета

```
<TextView  
...  
android:accessibilityHeading="true" />
```

Лістинг 8 - Встановлення елемента як заголовка в кодї

```
headingLabel.isAccessibilityHeading = true
```

Усі інтерактивні елементи, такі як кнопки, прапорці, перемикачі та інші елементи керування мають бути не менше 44 квадратних відносних пікселів, щоб відповідати вимогам доступності контенту. Android рекомендує використовувати мінімум 48 відносних пікселів. Android пропонує простий спосіб переконатися, що це завжди так. Це буде підтримуватися незалежно від пристрою чи решти ваших UI. Установіть значення `minWidth` і `minHeight` для будь-яких інтерактивних елементів керування (Лістинг 9 і Лістинг 10). Візьміть звичку робити це як

стандартну властивість для будь-якого елемента керування, який ви додаєте до свого інтерфейсу.

Лістинг 9 - Встановлення мінімального розміру кнопки зображення в XML

```
<ImageButton  
...  
android:minHeight="48dp"  
android:minWidth="48dp" />
```

Лістинг 10 - Встановлення мінімального розміру кнопки зображення в Kotlin

```
ImageButton.minWidth = 48  
ImageButton.minHeight = 48
```

Властивість `labelFor` встановлюється в `TextView`, який є «міткою» для елемента керування і надається як аргумент (Лістинг 11 та Лістинг 12). Слід забезпечити ідентифікатор представлення, до якого потрібно прив'язати цю мітку.

Лістинг 11 - Встановлення текстової мітки для елемента керування редагуванням тексту в XML

```

<LinearLayout
    ... >
    <TextView
        ...
        android:layout_marginBottom="0dp"
        android:labelFor="@+id/editText"
        android:text="@string/name" />
    <EditText
        ...
        android:id="@+id/editText"
        android:layout_marginTop="0dp"
        android:text="" />
</LinearLayout>

```

Лістинг 12 - Встановлення мітки для елемента керування EditText у Kotlin

```
textView.labelFor = editText.id
```

3.3 Порядок обходу дерева доступності

Дерево доступності будується в природному порядку читання. Це означає порядок що TalkBack та інші допоміжні технології перейдуть з верхнього від крайнього лівого до крайнього нижнього правого елемента. Це загалом правильна поведінка. Але деякі конструкції, такі як елементи в шаховому порядку, можуть читатися по-іншому ніж візуальний користувач може їх прочитати. Приклад такого макета може можна побачити в магазині Google Play, де вказані номери завантажень програми представлені прямо вертикально під заголовком «завантаження». У цій ситуації ми можемо захотіти покращити наше дерево доступності повідомляє Android, який елемент має бути наступним чи попереднім, коли служба доступності керує нашим інтерфейсом користувача. Android має два різні способи визначення порядку обходу залежно від типу вашої технології, яку клієнт використовує. Якщо ви знайдете порядок обходу, який

визначає Android, він не є ідеальним, тоді ми повинні бути впевнені, що ми встановили обидва ці параметри.

Будемо вважати, що у нас є три кнопки. Ми позначимо їх «1», «2» і «3» для наочності. Ми хочемо гарантувати, що ці елементи будуть переміщені в правильний порядок зростання числа. Для цього ми б встановили після і перед значеннями на 2 `accessibilityTraversalBefore` значення може бути 1, тому що ми хочемо, щоб 1 проходив до 2. Таким чином, це `accessibilityTraversalAfter` значення 3, оскільки ми хочемо, щоб 3 прийшло після 2. XML для його створення можна побачити в лістингу 12.

Лістинг 12 – Встановлення порядку обходження доступності в XML

```
<Button...
  android:id="@+id/button1"
  android:text="@string/one" />
<Button...
  android:id="@+id/button2"
  android:accessibilityTraversalAfter="@id/button3"
  android:accessibilityTraversalBefore="@id/button1"
  android:text="@string/two" />
<Button...
  android:id="@+id/button3"
  android:text="@string/three" />
```

Також слід встановити їх у коді за допомогою `setAccessibilityTraversal`. Методи `After()` і `setAccessibilityTraversalBefore()` і передача ідентифікатора перегляду, який ви хочете відвідати до або після, як у лістингу 13.

Лістинг 13 - Налаштування порядку обходу доступності в Kotlin

`two.setAccessibilityTraversalAfter(one.id)`

`two.setAccessibilityTraversalBefore(three.id)`

У порядку обходу доступності нам потрібно встановити порядок фокусування. Порядок фокусування використовується Android, коли користувачі переміщуються по інтерфейсу користувача за допомогою панелі напрямків, пульта дистанційного керування або клавіші зі стрілками на клавіатурі. Ми встановлюємо їх на елемент, який потрібно вказати на який елемент слід звернути увагу, коли клієнт натискає клавішу конкретний напрям. Властивості: `nextFocusDown`, `nextFocusUp`, `nextFocusLeft` та `nextFocusRight` для обробки введення клавіш зі стрілками та `nextFocusForward` для введення вкладки.

Щоб знати, як представити свій погляд користувачеві спеціальних можливостей, Android збирає інформацію про ваше представлення та його поточний статус. Це включає розмір і положення зору. Коли Android запитує цю інформацію, він викликає `onInitializeAccessibilityNodeInfo()` у вашому поданні, передаючи ініціалізований `AccessibilityNodeInfo` для заповнення (Лістинг 14). Ви завжди повинні викликати `super` тут, щоб дозволити Android заповнити щось розумне за замовчуванням. Потім ви можете додати будь-які додаткові атрибути, які потрібні вашому елементу керування. Це може включати в себе такі атрибути, як `isHeading` або `isChecked`.

Лістинг 14 - Додавання властивостей до інформації про доступність спеціального перегляду

```

class MyView: View {
    constructor(context: Context, attrs: AttributeSet):
        super(context, attrs)
        override fun onInitializeAccessibilityNodeInfo(info:
            AccessibilityNodeInfo) {
            super.onInitializeAccessibilityNodeInfo(info)
            Chapter 4 Android Accessibility Model
            // Accessibility services will treat your view as
            toggleable.
            info.isCheckable = true
        }
    }
}

```

Якщо ви вибираєте використовувати надані Android управління, але хотілося б більш чіткого контролю над тим, як працює доступність для кожного, то слід використовувати клас `AccessibilityDelegate`. Створіть клас, який розширюється `AccessibilityDelegate`; потім встановіть це як делегат у вашому представленні (Лістинг 15). Делегат може замінити ті самі методи доступності які були розглянуті раніше.

Лістинг 15 – Налаштування делегування доступності

```

class MyAccessibilityDelegate : View.AccessibilityDelegate() {
    override fun onInitializeAccessibilityNodeInfo(host: View,
        info: AccessibilityNodeInfo) {
        super.onInitializeAccessibilityNodeInfo(host, info)
        // add custom actions
    }
}

class MyActivity : Activity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.my_activity)
        val accessibilityDelegate = MyAccessibilityDelegate()
        button.accessibilityDelegate = accessibilityDelegate
    }
}

```

3.4 Семантичні погляди магазину Google Play

Створення семантичних уявлень – це потужна передова техніка для створення найбільш доступних додатків. Семантичний вигляд пов'язує два або більше елементи, які мають зв'язний сенс. Android потім представляє їх для користувачів доступності як один елемент. Це зменшує кількість шуму і робить навігацію по додатку набагато простішою, швидшою та менш негативною. Якщо ви дійсно прагнете надати своїм користувачам найкращу доступність слід використовувати цю техніку.

Google використовує цю техніку в магазині Google Play. Коли дивлячись на сторінку зі списком програми чи гри, ми бачимо три поля прямо під піктограмою програми (рисунок 7, сторінка 43) – рейтинг програми, кількість завантажень і вікове обмеження.

4.4 ★
1K reviews

100K+
Downloads

3
PEGI 3 ⓘ

Рисунок 7 – Інформація про додаток в Google Play Store

Перший пункт, рейтинг програми, відображає «4,4 *», а потім рядок «1 тис. відгуків». Крім того, текст рейтингу та вікове обмеження є кнопками. Якщо ви отримуєте доступ до цих елементів із увімкненим TalkBack без зміни в дереві доступності, це те, що озвучуватиметься: «Чотири зірки. Двічі торкніться, щоб активувати.» Проведіть пальцем. «100 тис.». Проведіть пальцем. «1 тисяча відгуків.» Проведіть пальцем. «Завантаження». Проведіть пальцем. «Вік 3 +. Двічі торкніться, щоб активувати».

Це багато свайпів. Одним змахом ми чуємо лише слово «завантаження». Що означає 100 тис. На щастя, TalkBack представляє це не так. Розробники Google Play Store покращили дерево доступності, щоб забезпечити кращий досвід. Натомість це взаємодія: «Середня оцінка чотири бали чотири зірки з тисячі оглядів. Двічі торкніться, щоб активувати.» Проведіть пальцем. «Завантажено понад 100 тисяч разів». Проведіть пальцем. «Рейтинг вмісту вище віку 3 років. Двічі торкніться, щоб активувати»

Групування цих елементів має ряд переваг. По-перше, менше потрібні свайпи, від п'яти до трьох. Це означає, що навігація швидша і простіше. Елементи, пов'язані з групуванням, як-от значення перегляду програми та кількість оглядів надає контекст для того, що означає кожне значення. Текст також було змінено, щоб бути більш значущим для клієнтів TalkBack змінюючи «100 тисяч» на «100 тисяч плюс». Хоча візуально це зрозуміло бачите, що К означає тисячу, ми рідко говоримо «к» вголос, щоб означати тисячу.

ВИСНОВКИ

В результаті виконаної кваліфікаційної роботи магістра на тему “Розробка мобільного додатку на базі Android для людей з інклюзією” було проаналізовано та описано актуальність обраної теми, оскільки відсоток людей з тимчасовою чи по життєвою інклюзією є достатньо великим, а кожен із нас сьогодні як в Україні так і у світі стикається із людьми з обмеженими можливостями. Проаналізувавши та зрозумівши проблеми цієї величезної групи людей, було обрано об’єкт дослідження, основними вимогами якого було встановлено доступність, інтуїтивна зрозумілість та простота. Було визначено основні види критерії, які повинні враховуватись при розробці будь якого програмного забезпечення для надання доступності користування цим програмним забезпеченням для людей з інклюзією.

Відповідно до особливостей такого програмного забезпечення та вимог, які слід притримуватись при розробці, було прийняте рішення скористатися операційною системою Android, яка є ще і платформою для мобільних пристроїв, які є доступними для кожного, що є надзвичайно важливим моментом для реалізації та поширення поставлених задач. Перевагами обраної операційної системи є відкритість, вільність доступу до зовнішніх носіїв та внутрішньої пам’яті, а також можливість зміни та налаштування інтерфейсу з урахуванням усіх критеріїв, що можуть негативно впливати на людей з інклюзією. Що є надзвичайно важливим при розробці додатку для людей із обмеженими можливостями.

Також було проаналізовано законодавства провідних країн та регіонів світу, а саме Сполучених штатів Америки та Європейського Союзу щодо передбачення надання доступності користування розроблюваного програмного забезпечення для людей з інклюзією. Після цього були визначені усі критерії для дотримання вимог цих законів при розробці власного мобільного додатку.

Після аналізу предметної області та визначення усіх проблем в обраній галузі було спроектовано архітектуру майбутнього додатку. А також безпосередньо розроблено, пройдено тестування за допомогою тест кейсів та впроваджено додаток на базі Android.

Відповідно до розробленого програмного забезпечення, а саме мобільного додатку на базі Android для людей з інклюзією було визначено, виконано та написано уся необхідна документація, усі пункти поставлені та сформовані у завданні на кваліфікаційну роботу, а саме підстави до розробки, призначення програмної системи, функціональні, технічні та програмні вимоги, етапи розробки, вся супровідна документація та порядок задач кваліфікаційної роботи. Також були розроблені необхідні документи до роботи, а саме пояснювальна записка, презентація, рецензія на проект та диск з проектом.

Згідно із виконаним проектом роблю висновок, що програмний додаток на операційній системі Android у повному обсязі забезпечує дотримання усіх вимог законодавства щодо надання доступності користування програмним забезпеченням людям з обмеженими можливостями, а також розроблений програмний продукт покриває та забезпечує усі потреби його користувачам.