Politechnika Warszawska

Projekt nr 3 TELM

E-karty gorączkowe dla oddziału szpitalnego z funkcją czytania kodów kreskowych i znaczników NFC w celu identyfikacji pacjentów

grupa: F
Karolina Trzcińska
305186
Filip Żarnowiec
297497

Prowadzący:

dr inż. Andrzej Rychter

Warszawa, 21.01.2024

Spis treści

Założenia projektowe

Aplikacja ma na celu wspomaganie identyfikacji pacjentów dla oddziału szpitalnego. E-karty są realizowane w formie systemu klient-serwer. Serwer o architekturze REST, za pomocą protokołu HTTP, komunikuje się z aplikacją mobilną na system Android. Za pomocą aplikacji lekarz może zidentyfikować pacjenta za pomocą kodu kreskowego lub taga NFC. Interfejs umożliwia obejrzeć wykres zarejestrowanej temperatury pacjenta oraz inne badania. W związku z tym, że aplikacja przetwarza dane medyczne, system zapewnia bezpieczeństwo komunikacji oraz dostępu do danych za pomocą standardu JSON Web Tokens.

Wymagania

- serwer REST odpowiadający za przechowywanie i bezpieczne udostępnianie danych pacjenta
- aplikacja kliencka na system operacyjny Android
    - wyświetlanie danych o pacjencie
    - czytnik kodów kreskowych
    - czytnik tagów NFC
- bezpieczeństwo aplikacji oraz danych medycznych

Wykorzystane narzędzia

Zarówno aplikacja mobilna i serwer została zaimplementowana w języku Kotlin. Jest to rekomendowany język do pisania aplikacji na system operacyjny Android. Do realizacji interfejsu użytkownika użyta została biblioteka Jetpack Compose, która pozwala na deklaratywne zarządzanie elementami widoku. Design aplikacji oraz gotowe komponenty pochodzą z biblioteki Material Design 3.
Serwer powstał w frameworku Ktor, który ma architekturę wtyczkową. Do uwierzytelniania, trasowania oraz obsługi bazy danych zastosowane zostały odpowiednie wtyczki. Do przechowywania danych oddziału szpitalnego wykorzystane została baza danych H2.

Przechowuje ona dane w pamięci RAM serwera, więc w rzeczywistym zastosowaniu nie jest to rozwiązanie skalowalne, ale dobrze nadaje się w tym projekcie, ponieważ nie wymaga dodatkowej konfiguracji na maszynie gospodarza.

Architektura

W implementacji aplikacji zastosowany został wzorzec projektowy Model - Widok - Model Widoku (z ang. MVVM - Model-View-ViewModel). W konsekwencji aplikacja została podzielona na trzy warstwy, z których model zrealizowany jest poprzez serwer REST, a po stronie aplikacji klienckiej jest warstwa widoku, która odpowiadania za interakcje użytkownika z systemem. W celu modularyzacji logika działania aplikacji jest przeniesiona do warstwy modelu widoku, który zarządza stanem aplikacji oraz synchronizacją danych z serwerem.

Model

Struktury danych

Struktury danych będa w postaci tablic. Poniżej zaprezentowane są wizualizacje tych tablic, z rodzajem danych, które będą one zawierały.
Struktura danych dla użytkownika.

USER_ID	NAME	SURNAME	ID_DOCTOR	ROLE	LOGIN	PASSWORD
Dane typu UUID			ID użytkownika - lekarza	lekarz / pacjent		

Struktura danych dla poszczególnych badań.

- Dla pomiaru temperatury:

USER_ID	DETAIL_DATE	MEASURE_VALUE	MEASURE_ID
Dane typu UUID	Zmienna typu long	Dane typu double	Dane typu UUID

- Dla pomiaru EKG:

USER_ID	DETAIL_DATE	ARRAY_OF_VALUE	MEASURE_ID
Dane typu UUID	Zmienna typu long		Dane typu UUID

- Dla morfologii:

USER_ID	DETAIL_DATE	MEASURE_VALUE	MEASURE_ID	HT_VALUE	HB_VALUE	MCV_VALUE	MCHC_VALUE
Dane typu UUID	Zmienna typu long	Dane typu double	Dane typu UUID	Wartość Hematokrytu	Wartość Hemoglobiny	Wartość MCV (czyli objętość krwinki)	Wartość MCHC (stężenie hemoglobiny w krwince)

Serwer REST

- ***/login***
  Endpoint odpowiedzialny za walidację otrzymanych danych logowania. W przypadku sukcesu, zostaje generowany token JWT, który zostaje zwrócony do klienta. Będzie on służył do zabezpieczenia komunikacji oraz kontroli dostępu do zasobów serwera.

  - Parametry:
    username: string - nazwa użytkownika
    password: string - hasło dostępu

  - Odpowiedź
    user: json - dane personalne o użytkowniku
    token: string - wygenerowany token jwt

- **/*patient*/{id}/*info***
  Dostęp do tej “końcówki” ma dostęp tylko osoba, która jest właścicielem danych lub jest pracownikiem oddziału. Zwraca wszystkie badania dla użytkownika o podanym identyfikatorze w postaci json.

  - Parametry:
    id: string - identyfikator użytkownika
    token: string

  - Odpowiedź
    info: json - dane o użytkowniku oraz wyniki jego badań

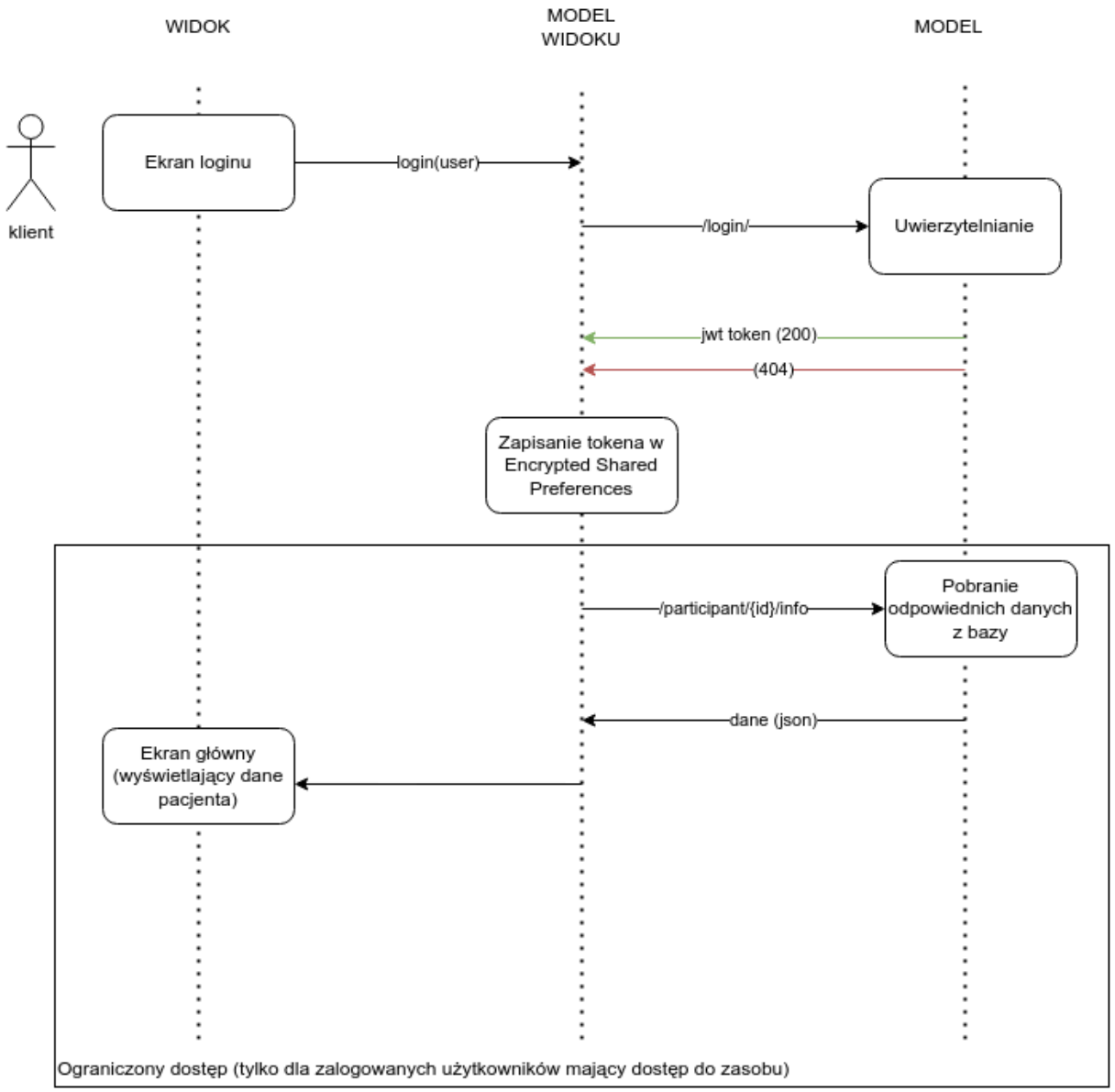
Widok

propozycja widoku zawarta w testach akceptacyjnych

Model widoku

Kontekst (zasięg) modelu widoku przechowuje stan aplikacji, na której podstawie rysuje się widok. Wykorzystany został wzorzec jednokierunkowego przepływu danych, to znaczy, że wydarzenie wygenerowane przez użytkownika zostaje rozpatrzone przez warstwę VM,

a następnie, jeżeli akcja zmieniła stan, interfejs użytkownika zostaje przebudowany. Przepływ danych w aplikacji podczas logowania został przedstawiony na rysunku 1.



Rysunek 1: Scenariusz przepływu danych w aplikacji podczas logowania i dostępu do danych pacjenta.

Instrukcja instalacji

Projekt jest budowany za pomocą narzędzia gradle. W celu testowania aplikacji w sieci lokalnej należy ustawić stałą DOMAIN w pliku composeApp/src/androidMain/kotlin/pl/pw/ekartapacjenta/logic/NetworkManager.kt na odpowiadającej domenie lub adresie serwera. Aplikację mobilną zbudować używając skryptu gradlew, wykonując następującą komendę: *./gradlew composeApp:build*. Zbudowana aplikacja będzie w folderze bulld/output. Serwer można uruchomić za pomocą komendy *./gradlew serwer:run.*

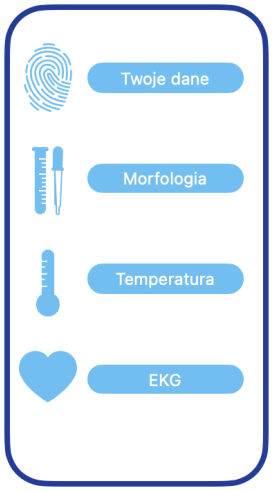
Dane testowe:

- pacjent -> login: testPatient hasło: 123
- lekarz -> login: testDoctor hasło: 123

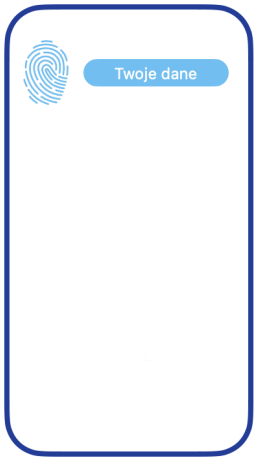
Testy akceptacyjne (Instrukcja użytkownika)

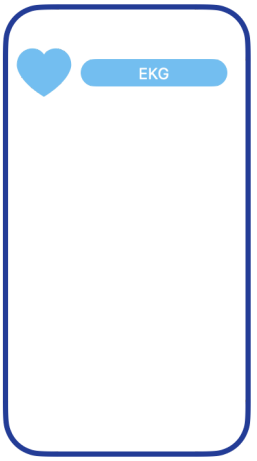
Test akceptacyjny z poziomu pacjenta.

Po wybraniu aplikacji z ekranu telefonu otwiera się ekran z logowaniem do aplikacji. Jest to okno z polami do uzupełnienia loginu i hasła w celu zalogowania się do aplikacji w celu przeglądania danych w niej zawartych.
Jeśli wprowadzony zostanie błędny login lub hasło użytkownik nie będzie mógł dostać się do konta w aplikacji.
Jeśli użytkownik wprowadzi poprawne login i hasło zostanie przeniesiony on do swojego ekranu początkowego swojego konta.



Znajdując się w ekranie początkowym użytkownikowi otwiera się menu, z którego może dostać się do poszczególnych kart z konkretnym typem badań: morfologią, pomiarem temperatury oraz wynikami pomiaru EKG.
Chcąc uzyskać dostęp do konkretnej karty z danymi użytkownik musi kliknąć w wybrany przycisk obok ikony.
Zostaje on przekierowany do karty z danymi badaniami.



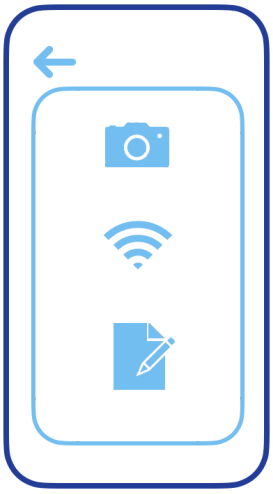


Nazwa danej karty widnieje u góry ekranu, a na poniżej będa znajdywać się wyniki badań.

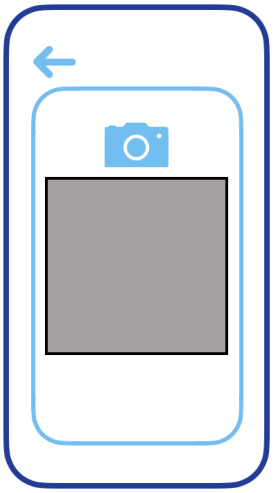
Test akceptacyjny z poziomu lekarza.

Po wybraniu aplikacji z ekranu telefonu otwiera się ekran z logowaniem do aplikacji tak samo jak w przypadku logowania przez pacjenta.

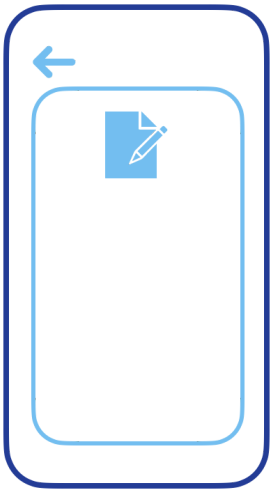
Po zalogowaniu się lekarz będzie miał możliwość wyszukania konta pacjenta po jego nazwisku i imieniu, które będzie mógł wpisać ręcznie klikając na pole w środkowej części ekranu.
Lekarz będzie mógł również wyszukać karty pacjenta z pomocą skanera kodu kreskowego lub czytnika NFC.
W tym celu użytkownik musi kliknąć okrągły “unoszący się” przycisk (floating button) z plusem w prawym dolnym rogu. Po jego naciśnięciu otworzy się dodatkowe menu dostępne tylko dla profilu lekarza.



W menu znajdują się opcje: włączenia aparatu w celu zeskanowania kodu kreskowego pacjenta, w celu szybszej identyfikacji, aktywacja czytnika NFC również służąca do szybkiego znalezienia karty pacjenta oraz ostatnia ikona umożliwiająca dodanie danych medycznych, dodanie opisu lub edycję już istniejącego tekstu.



Po kliknięciu pierwszej ikony otworzy się okno, które trzeba będzie skierować na kod kreskowy w celu znalezienia karty konkretnego pacjenta. Druga ikona służy do znalezienia karty pacjenta za pomocą czytnika NFC.



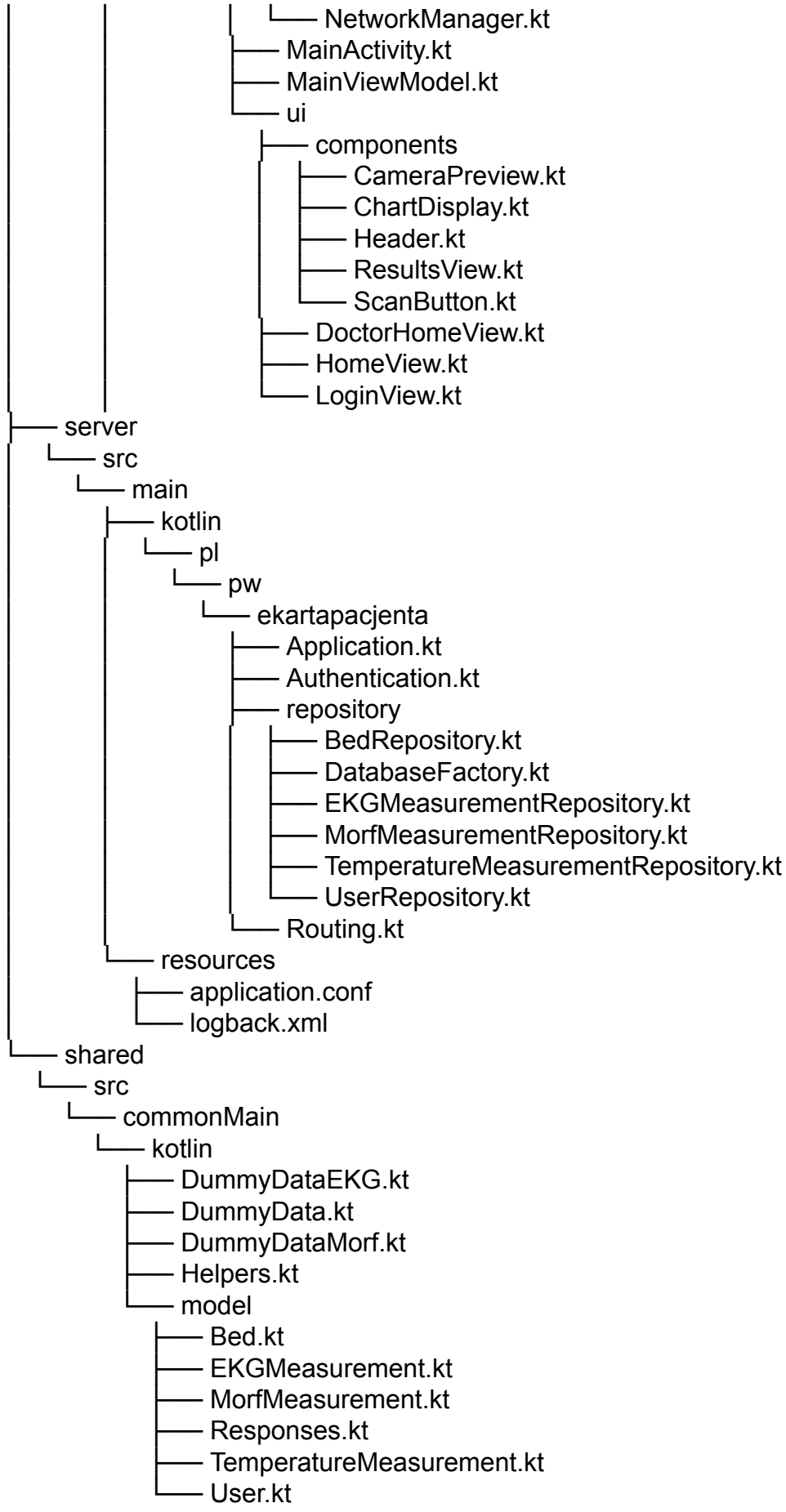
Trzecia ikona służy do umożliwienia lekarzowi wprowadzenia nowych lub edycji już istniejących danych, która znajdzie swoje zastosowanie już po wejściu w kartę konkretnego pacjenta.
Floating button będzie widoczny na każdym ekranie w aplikacji z poziomu lekarza, nie tylko na początku. Zatem lekarz może wprowadzić dane w każdym momencie przeglądania kart.

Podział pracy

Karolina Trzcińska była odpowiedzialna za warstwę widoku aplikacji, a warstwa modelu widoku została napisana wspólnie. Filip Żarnowiec zaimplementował serwer.

Spis plików źródłowych

```
composeApp
    src
        androidMain
            AndroidManifest.xml
            kotlin
                pl
                    pw
                        ekartapacjenta
                            App.kt
                            logic
                                EncryptedSharedPref.kt
```


NetworkManager.kt
MainActivity.kt
MainViewModel.kt
ui
components
CameraPreview.kt
ChartDisplay.kt
Header.kt
ResultsView.kt
ScanButton.kt
DoctorHomeView.kt
HomeView.kt
LoginView.kt
server
src
main
kotlin
pl
pw
ekartapacjenta
Application.kt
Authentication.kt
repository
BedRepository.kt
DatabaseFactory.kt
EKGMeasurementRepository.kt
MorfMeasurementRepository.kt
TemperatureMeasurementRepository.kt
UserRepository.kt
Routing.kt
resources
application.conf
logback.xml
shared
src
commonMain
kotlin
DummyDataEKG.kt
DummyData.kt
DummyDataMorf.kt
Helpers.kt
model
Bed.kt
EKGMeasurement.kt
MorfMeasurement.kt
Responses.kt
TemperatureMeasurement.kt
User.kt

Użyte biblioteki

Biblioteki podane w formacie *id_grupy:id_artefaktu:wersja*

- androidx.activity:activity-compose:1.8.2
- androidx.activity:activity-ktx:1.8.2
- androidx.activity:activity:1.8.2
- androidx.annotation:annotation-experimental:1.3.0
- androidx.annotation:annotation-experimental:1.3.1
- androidx.annotation:annotation-jvm:1.7.0
- androidx.annotation:annotation:1.5.0
- androidx.arch.core:core-common:2.2.0
- androidx.arch.core:core-runtime:2.2.0
- androidx.camera:camera-camera2:1.3.0
- androidx.camera:camera-core:1.3.0
- androidx.camera:camera-extensions:1.3.0
- androidx.camera:camera-lifecycle:1.3.0
- androidx.camera:camera-video:1.3.0
- androidx.camera:camera-view:1.3.0
- androidx.collection:collection:1.0.0
- androidx.collection:collection:1.2.0
- androidx.compose.animation:animation-android:1.5.4
- androidx.compose.animation:animation-android:1.6.0-alpha06
- androidx.compose.animation:animation-core-android:1.5.4
- androidx.compose.animation:animation-core-android:1.6.0-alpha06
- androidx.compose.foundation:foundation-android:1.5.4
- androidx.compose.foundation:foundation-android:1.6.0-alpha06
- androidx.compose.foundation:foundation-layout-android:1.5.4
- androidx.compose.foundation:foundation-layout-android:1.6.0-alpha06
- androidx.compose.material:material-android:1.5.4
- androidx.compose.material:material-icons-core-android:1.5.4
- androidx.compose.material:material-ripple-android:1.5.4
- androidx.compose.runtime:runtime-android:1.5.4
- androidx.compose.runtime:runtime-android:1.6.0-alpha06
- androidx.compose.runtime:runtime-saveable-android:1.5.4
- androidx.compose.runtime:runtime-saveable-android:1.6.0-alpha06
- androidx.compose.ui:ui-android:1.5.4
- androidx.compose.ui:ui-android:1.6.0-alpha06
- androidx.compose.ui:ui-geometry-android:1.5.4
- androidx.compose.ui:ui-geometry-android:1.6.0-alpha06
- androidx.compose.ui:ui-graphics-android:1.5.4
- androidx.compose.ui:ui-graphics-android:1.6.0-alpha06
- androidx.compose.ui:ui-text-android:1.5.4
- androidx.compose.ui:ui-text-android:1.6.0-alpha06
- androidx.compose.ui:ui-tooling-android:1.5.4
- androidx.compose.ui:ui-tooling-android:1.6.0-alpha06
- androidx.compose.ui:ui-tooling-data-android:1.5.4

- androidx.compose.ui:ui-tooling-data-android:1.6.0-alpha06
- androidx.compose.ui:ui-tooling-preview-android:1.5.4
- androidx.compose.ui:ui-tooling-preview-android:1.6.0-alpha06
- androidx.compose.ui:ui-unit-android:1.5.4
- androidx.compose.ui:ui-unit-android:1.6.0-alpha06
- androidx.core:core-ktx:1.11.0-beta02
- androidx.core:core-ktx:1.9.0
- androidx.core:core:1.11.0-beta02
- androidx.core:core:1.9.0
- androidx.lifecycle:lifecycle-common-java8:2.7.0
- androidx.lifecycle:lifecycle-common:2.7.0
- androidx.lifecycle:lifecycle-livedata-core-ktx:2.7.0
- androidx.lifecycle:lifecycle-livedata-core:2.7.0
- androidx.lifecycle:lifecycle-livedata:2.7.0
- androidx.lifecycle:lifecycle-runtime-ktx:2.7.0
- androidx.lifecycle:lifecycle-runtime:2.7.0
- androidx.lifecycle:lifecycle-viewmodel-compose:2.7.0
- androidx.lifecycle:lifecycle-viewmodel-ktx:2.7.0
- androidx.lifecycle:lifecycle-viewmodel-savedstate:2.7.0
- androidx.lifecycle:lifecycle-viewmodel:2.7.0
- androidx.savedstate:savedstate-ktx:1.2.1
- androidx.savedstate:savedstate:1.2.1
- androidx.security:security-crypto:1.0.0
- androidx.versionedparcelable:versionedparcelable:1.1.1
- ch.qos.logback:logback-classic:1.4.13
- ch.qos.logback:logback-core:1.4.13
- ch.randelshofer:fastdoubleparser:0.8.0
- com.auth0:java-jwt:4.4.0
- com.auth0:jwks-rsa:0.22.1
- com.fasterxml.jackson.core:jackson-annotations:2.15.0
- com.fasterxml.jackson.core:jackson-core:2.15.0
- com.fasterxml.jackson.core:jackson-databind:2.15.0
- com.github.PhilJay:MPAndroidChart:v3.1.0
- com.google.accompanist:accompanist-permissions:0.33.2-alpha
- com.google.code.findbugs:jsr305:3.0.2
- com.google.errorprone:error_prone_annotations:2.18.0
- com.google.guava:failureaccess:1.0.1
- com.google.guava:guava:32.1.1-jre
- com.google.guava:listenablefuture:1.0
- com.google.zxing:core:3.3.0
- com.h2database:h2:2.1.214
- com.squareup.okhttp3:okhttp:4.12.0
- com.squareup.okio:okio-jvm:3.6.0
- com.typesafe:config:1.4.2
- commons-codec:commons-codec:1.11
- commons-logging:commons-logging:1.2
- io.ktor:ktor-client-apache-jvm:2.3.6
- io.ktor:ktor-client-cio-jvm:2.3.6

- io.ktor:ktor-client-content-negotiation-jvm:2.3.6
- io.ktor:ktor-client-core-jvm:2.3.6
- io.ktor:ktor-client-okhttp-jvm:2.3.6
- io.ktor:ktor-events-jvm:2.3.6
- io.ktor:ktor-http-cio-jvm:2.3.6
- io.ktor:ktor-http-jvm:2.3.6
- io.ktor:ktor-io-jvm:2.3.6
- io.ktor:ktor-network-jvm:2.3.6
- io.ktor:ktor-network-tls-certificates-jvm:2.3.6
- io.ktor:ktor-network-tls-jvm:2.3.6
- io.ktor:ktor-serialization-jvm:2.3.6
- io.ktor:ktor-serialization-kotlinx-json-jvm:2.3.6
- io.ktor:ktor-serialization-kotlinx-jvm:2.3.6
- io.ktor:ktor-server-auth-jvm:2.3.6
- io.ktor:ktor-server-auth-jwt-jvm:2.3.6
- io.ktor:ktor-server-auto-head-response-jvm:2.3.6
- io.ktor:ktor-server-caching-headers-jvm:2.3.6
- io.ktor:ktor-server-call-id-jvm:2.3.6
- io.ktor:ktor-server-call-logging-jvm:2.3.6
- io.ktor:ktor-server-compression-jvm:2.3.6
- io.ktor:ktor-server-conditional-headers-jvm:2.3.6
- io.ktor:ktor-server-content-negotiation-jvm:2.3.6
- io.ktor:ktor-server-core-jvm:2.3.6
- io.ktor:ktor-server-cors-jvm:2.3.6
- io.ktor:ktor-server-data-conversion-jvm:2.3.6
- io.ktor:ktor-server-default-headers-jvm:2.3.6
- io.ktor:ktor-server-double-receive-jvm:2.3.6
- io.ktor:ktor-server-forwarded-header-jvm:2.3.6
- io.ktor:ktor-server-host-common-jvm:2.3.6
- io.ktor:ktor-server-hsts-jvm:2.3.6
- io.ktor:ktor-server-http-redirect-jvm:2.3.6
- io.ktor:ktor-server-jvm:2.3.6
- io.ktor:ktor-server-method-override-jvm:2.3.6
- io.ktor:ktor-server-netty-jvm:2.3.6
- io.ktor:ktor-server-partial-content-jvm:2.3.6
- io.ktor:ktor-server-rate-limit-jvm:2.3.6
- io.ktor:ktor-server-sessions-jvm:2.3.6
- io.ktor:ktor-server-status-pages-jvm:2.3.6
- io.ktor:ktor-server-test-host-jvm:2.3.6
- io.ktor:ktor-server-tests-jvm:2.3.6
- io.ktor:ktor-server-websockets-jvm:2.3.6
- io.ktor:ktor-test-dispatcher-jvm:2.3.6
- io.ktor:ktor-utils-jvm:2.3.6
- io.ktor:ktor-websocket-serialization-jvm:2.3.6
- io.ktor:ktor-websockets-jvm:2.3.6
- io.netty:netty-buffer:4.1.97.Final
- io.netty:netty-codec-http2:4.1.97.Final
- io.netty:netty-codec-http:4.1.97.Final

- io.netty:netty-codec:4.1.97.Final
- io.netty:netty-common:4.1.97.Final
- io.netty:netty-handler:4.1.97.Final
- io.netty:netty-resolver:4.1.97.Final
- io.netty:netty-transport-classes-epoll:4.1.97.Final
- io.netty:netty-transport-classes-kqueue:4.1.97.Final
- io.netty:netty-transport-native-epoll:4.1.97.Final
- io.netty:netty-transport-native-kqueue:4.1.97.Final
- io.netty:netty-transport-native-unix-common:4.1.97.Final
- io.netty:netty-transport:4.1.97.Final
- junit:junit:4.13.2
- me.dm7.barcodescanner:core:1.9.8
- me.dm7.barcodescanner:zxing:1.9.8
- net.bytebuddy:byte-buddy-agent:1.10.9
- net.bytebuddy:byte-buddy:1.10.9
- net.java.dev.jna:jna-platform:5.9.0
- net.java.dev.jna:jna:5.9.0
- org.apache.httpcomponents:httpasyncclient:4.1.5
- org.apache.httpcomponents:httpclient:4.5.13
- org.apache.httpcomponents:httpcore-nio:4.4.15
- org.apache.httpcomponents:httpcore:4.4.15
- org.checkerframework:checker-qual:3.33.0
- org.eclipse.jetty.alpn:alpn-api:1.1.3.v20160715
- org.fusesource.jansi:jansi:2.4.0
- org.hamcrest:hamcrest-core:1.3
- org.jetbrains.compose.components:library-android:1.5.11
- org.jetbrains.exposed:exposed-core:0.41.1
- org.jetbrains.exposed:exposed-dao:0.41.1
- org.jetbrains.exposed:exposed-jdbc:0.41.1
- org.jetbrains.kotlin:kotlin-reflect:1.8.22
- org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.8.22
- org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.9.10
- org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.8.22
- org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.9.10
- org.jetbrains.kotlin:kotlin-stdlib:1.9.21
- org.jetbrains.kotlin:kotlin-test-junit:1.9.21
- org.jetbrains.kotlin:kotlin-test:1.9.21
- org.jetbrains.kotlinx:kotlinx-coroutines-android:1.7.1
- org.jetbrains.kotlinx:kotlinx-coroutines-core-jvm:1.7.1
- org.jetbrains.kotlinx:kotlinx-coroutines-debug:1.7.1
- org.jetbrains.kotlinx:kotlinx-coroutines-jdk8:1.7.1
- org.jetbrains.kotlinx:kotlinx-coroutines-slf4j:1.7.1
- org.jetbrains.kotlinx:kotlinx-datetime-jvm:0.5.0
- org.jetbrains.kotlinx:kotlinx-serialization-core-jvm:1.5.1
- org.jetbrains.kotlinx:kotlinx-serialization-core-jvm:1.6.2
- org.jetbrains.kotlinx:kotlinx-serialization-json-jvm:1.5.1
- org.jetbrains.kotlinx:kotlinx-serialization-json-jvm:1.6.2
- org.jetbrains:annotations:23.0.0

- org.slf4j:slf4j-api:1.7.36
- org.slf4j:slf4j-api:2.0.7
- io.ktor:ktor-client-core-jvm:2.3.6
- io.ktor:ktor-events-jvm:2.3.6
- io.ktor:ktor-http-jvm:2.3.6
- io.ktor:ktor-io-jvm:2.3.6
- io.ktor:ktor-serialization-jvm:2.3.6
- io.ktor:ktor-utils-jvm:2.3.6
- io.ktor:ktor-websocket-serialization-jvm:2.3.6
- io.ktor:ktor-websockets-jvm:2.3.6
- org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.8.22
- org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.8.22
- org.jetbrains.kotlin:kotlin-stdlib:1.9.21
- org.jetbrains.kotlinx:kotlinx-coroutines-core-jvm:1.7.1
- org.jetbrains.kotlinx:kotlinx-coroutines-jdk8:1.7.1
- org.jetbrains.kotlinx:kotlinx-datetime-jvm:0.5.0
- org.jetbrains.kotlinx:kotlinx-serialization-core-jvm:1.6.2
- org.jetbrains.kotlinx:kotlinx-serialization-json-jvm:1.6.2
- org.jetbrains:annotations:13.0
- org.jetbrains:annotations:23.0.0
- org.slf4j:slf4j-api:1.7.36