



Animedex

Corso ITS Software Architect Specialist - Sez. A

Mark Andro Guevarra

Indice

1. Introduzione

- 1.1 Scopo del documento
- 1.2 Obiettivo personale
- 1.2 Obiettivo del progetto
- 1.3 Risorse esterne
- 1.4 Panoramica del prodotto

2. Descrizione generale

- 2.1 Tipologia prodotto
- 2.2 Funzionalità del prodotto
- 2.3 Caratteristiche degli utenti
- 2.4 Vincoli generali

3. Requisiti specifici

- 3.1 Requisiti funzionali
- 3.2 Requisiti non funzionali
- 3.3 Requisiti di interfaccia

4. Diagramma e descrizione dei casi d'uso

- 4.1 Caso d'uso dell'utente

5. API

- 5.1 Tabella endpoint

6. Modello E-R

7. Wireframe

8. Design

9. Architettura del sistema

Introduzione

Il seguente capitolo descrive generalmente le informazioni volte al chiarimento e guida per la lettura dell'intero documento

1.1 Scopo del documento

Questo documento ha il fine di descrivere le varie fasi per la creazione dell'applicazione android "Animedex": una ricerca iniziale per comprendere meglio il client-target, definizione dei requisiti funzionali e non funzionali, wireframe e design, l'utilizzo da parte degli attori/utilizzatori dell'app e l'architettura ideata

1.2 Obiettivo personale

Il mio obiettivo principale è quello di riuscire a creare un sistema funzionante, utilizzare i vari strumenti di lavoro per costruire una applicazione mobile Android, in particolare l'utilizzo degli API e la definizione e creazione di questi.

Migliorare le competenze per quanto riguarda il back end per poter consegnare informazioni e servizi alla mia piattaforma.

Capire meglio il percorso dei dati nella parte più alta del modello TCP/IP e la comunicazione tra un client e server.

Ho utilizzato Node.js con Express per creare il server, una tecnologia nuova per me, con l'obiettivo di apprendere, ma anche perché mi è stato consigliato da un compagno.

Ho fatto questa applicazione in modo tale di poter imparare mentre sto costruendo un progetto che mi piaccia, ho sfruttato questa specie di simulazione per poter ripassare ulteriormente gli argomenti affrontati.

1.3 Obiettivo del progetto

L'obiettivo dell'applicazione è di offrire all'utente una applicazione mobile personalizzata grazie a metodi implementati definite nel documento, oltre alle funzioni base, che è quello di una presentazione display degli anime disponibili, rendendo l'esperienza di consultazione semplice e intuitiva sulle piattaforme mobili.

1.4 Risorse esterne utilizzate

- Grafici Online e notizie riguardanti anime
- API Jikan

1.5 Panoramica del documento

Il documento è organizzato in varie sezioni suddivise per ogni fase della creazione del progetto Animedex: descrizione del prodotto, wireframe e design, requisiti specifici, casi d'uso e l'architettura del sistema

Descrizione generale

Il progetto consiste nella realizzazione di un'applicazione mobile nativa per Android, sviluppato per tutti gli appassionati di animazione giapponese (anime).

L'obiettivo principale è: offrire un'enciclopedia di anime e fornire agli utenti uno strumento semplice per monitorare i contenuti che hanno già visto.

2.1 Tipologia prodotto

Il sistema è un'applicazione mobile destinata per tutti i dispositivi Android

2.2 Funzionalità del prodotto

- La registrazione degli utenti
- L'autenticazione degli utenti
- La consultazione di una lista anime (serie animata)
- La consultazione dettagliata di ogni anime
- Il salvataggio di un anime alla propria area personale al completamento
- Il tracciamento delle ore spese
- Il tracciamento del numero di serie completate

2.3 Caratteristiche degli utenti

- L'utente registrato che può consultare la lista degli anime, aggiungere le serie completate nella dashboard e consultare la propria statistica personale

2.4 Vincoli generali

- La necessità dell'applicazione di essere accessibile sui dispositivi mobile esclusivamente ANDROID
- La necessità della registrazione per l'utilizzo del servizio
-

Requisiti Specifici

Questa parte affronteremo i requisiti funzionali e i requisiti non funzionali del sistema

3.1 Requisiti funzionali

- L'utente deve potersi registrare
- L'utente deve potersi autenticare
- L'utente ha accesso alla consultazione della lista degli anime
- L'utente è permesso di salvare la serie completata nel dashboard personale
- L'utente ha la possibilità di consultare la lista degli anime salvati
- L'utente deve vedere il tracciamento personale del tempo speso, serie completate e totali di episodi visti nella propria dashboard

3.2 Requisiti non funzionali

- Il sistema deve garantire **usabilità, sicurezza**, fornendo un'interfaccia intuitiva, protezione dei dati.
- Il sistema deve essere compatibile con dispositivi Android
- Il sistema deve garantire la **compatibilità con almeno il 45% dei dispositivi Android in circolazione**, basandosi sulle statistiche di distribuzione.

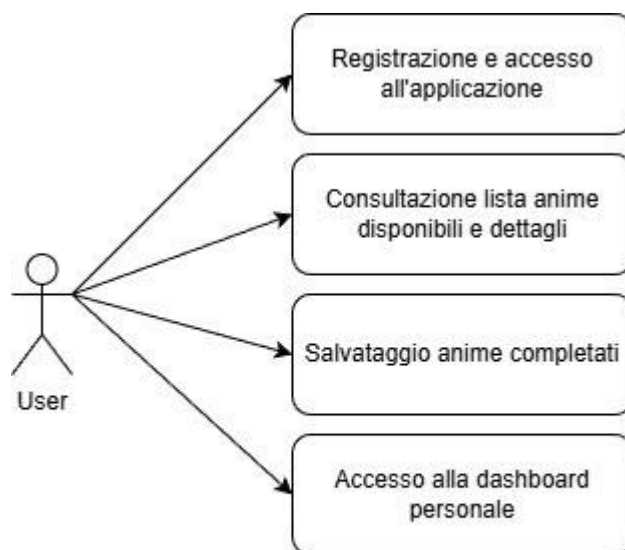
3.3 Requisiti di interfaccia

- L'interfaccia deve essere intuitiva e semplice

Diagramma e descrizione dei casi d'uso

Qui sarà l'illustrazione dei vari casi d'uso dell'applicazione mobile, attraverso illustrazioni e descrizioni dettagliate per descrivere come l'utente può utilizzare l'applicazione una volta registrati e fatto l'accesso.

4.1 Diagramma e caso d'uso dell'utente



1.Caso d'uso: **registrazione e accesso all'applicazione**

- Attore: User
- Precondizione: l'utente ha scaricato l'applicazione sul dispositivo (controllare la versione supportata)
- Sequenza degli eventi:
 - L'utente apre l'applicazione
 - L'utente si registra

- L'utente accede all'applicazione

2. Caso d'uso: **Consultazione lista anime e dettaglio**

- Attore: User
- Precondizione: l'utente ha fatto la registrazione e ha fatto l'autenticazione
- Sequenza degli eventi:
 - L'utente consulta la lista degli anime disponibili
 - L'utente sceglie e seleziona un anime da approfondire
 - L'utente accede all'area dettaglio dell'anime

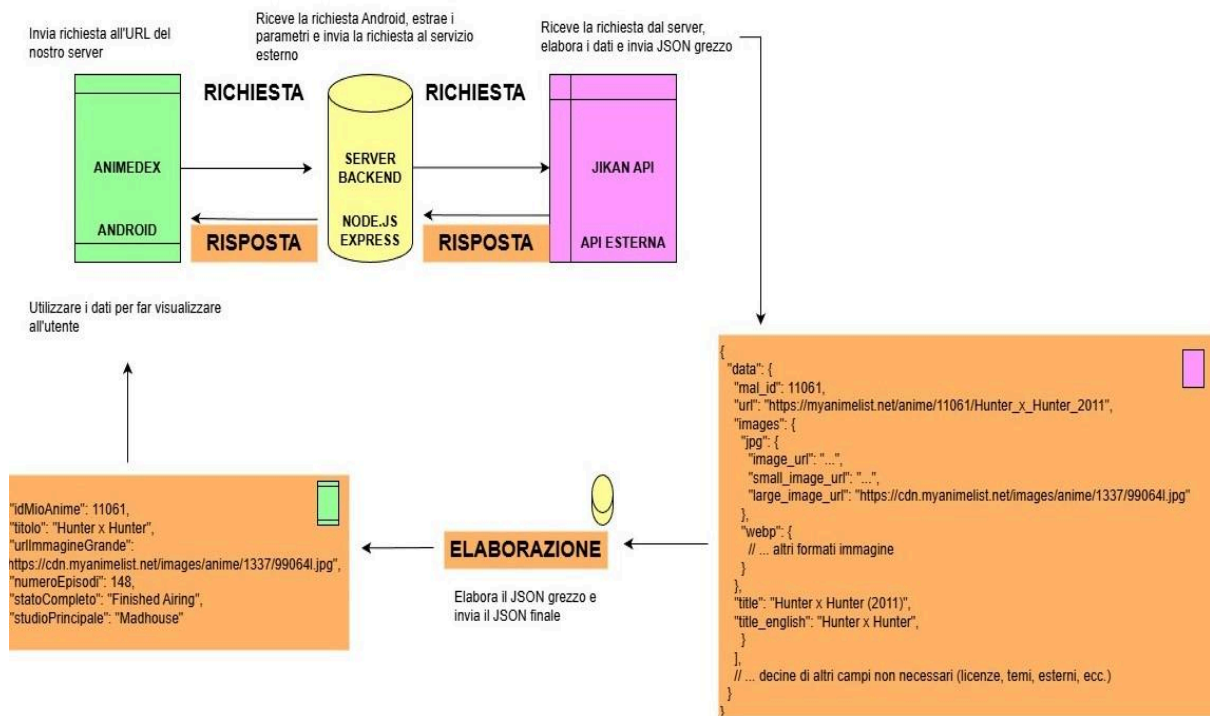
3. Caso d'uso: **Salvataggio anime completati**

- Attore: User
- Precondizione: l'utente è registrato e ha fatto l'autenticazione
- Sequenza degli eventi:
 - L'utente legge il dettaglio anime che ha selezionato
 - L'utente clicca il bottone "completato"

4. Caso d'uso: **Accesso alla dashboard personale**

- Attore: User
- Precondizione: l'utente ha fatto la registrazione e ha fatto l'autenticazione
- Sequenza degli eventi:
 - L'utente accede alla dashboard personale
 - L'utente vede le serie anime completate
 - L'utente consulta la propria statistica personale

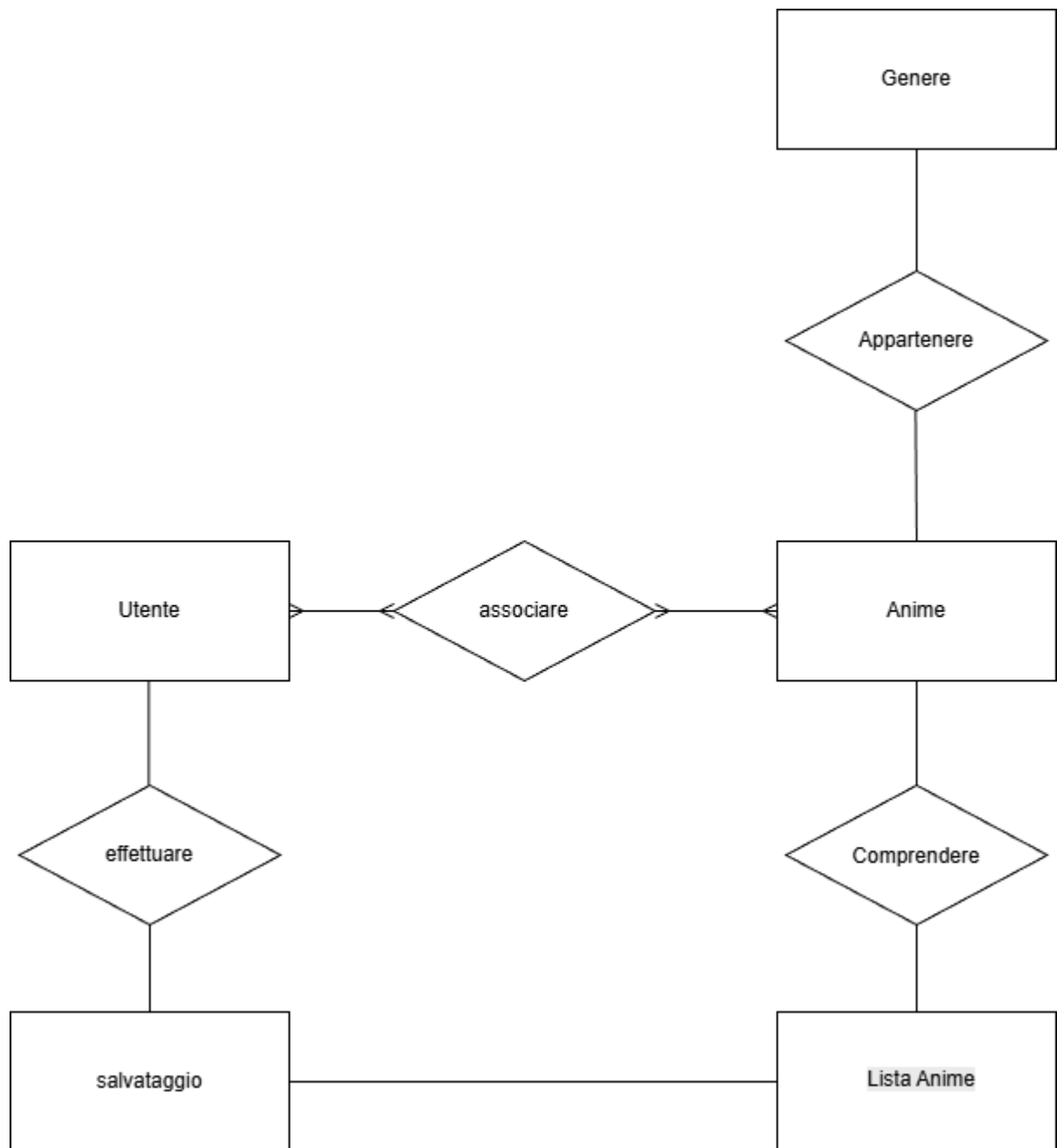
API



METODO	ENDPOINT	PARAMETRI	RISPOSTA	NOTE	ESEMP
GET	/anime	obbligatori: - opzionali: page (default: 1) limit (default: 25)	JSON con{ success, data: [[id,title,image,score]], pagination }	Restituisce lista anime per punteggio. Sorgente dati https://api.jikan.moe/v4/anime	GET http://localhost:3000/anime?page=1&limit=25
GET	/anime/search	obbligatori: q (parola/e da cercare)	JSON con { success, data: [[id,title,image,score]] }	Se non ci fossero parametri passati ci restituisce errore 400. Sorgente dati https://api.jikan.moe/v4/anime	GET http://localhost:3000/anime/search?q=Naruto
GET	/anime/:id	obbligatori: id(identificazione di un anime specifico con il suo id)	JSON con { id, title, image, score, synopsis, year, status }	Al click di una risorsa ottiene i dati completi (dettaglio) di un anime tramite Jikan https://api.jikan.moe/v4/anime/:id	GET http://localhost:3000/anime/20
POST	/register	obbligatori: nel body JSON {"username": "username", "password": "password" }	JSON con { message, user { id, username} }	Registrazione user	POST http://localhost:3000/api/auth/register BODY: { "username": "Baba", "password": "1234" }
POST	/login	JSON body: - username (string, obbligatorio) - password (string, obbligatorio)	JSON con {message, user: { id, username} }	Verifica che l'username e la password siano presenti; usa bcrypt per confrontare le password	POST http://localhost:3000/api/auth/login { "username": "mario", "password": "1234" }

POST	/completed	obbligatorie: nel body JSON {"user_id","anime_id","title"}	JSON con { message, anime{anime_id,user_id,title} }	Salvataggio anime completato nel database	POST http://localhost:3000/api/completed
POST	/completed/list	obbligatorie: nel body JSON {"user_id"}	JSON con { message, completed }	Recupero della lista di anime che lo user ha salvato nei completati	POST http://localhost:3000/api/completed/list

Modello E-R



1. Utente

- ID_Utente (PK)
- Username
- Fullname
- Email
- Password

2. Anime

- ID_Anime (PK)
- Titolo
- Descrizione
- Genere
- Numero_Episodi

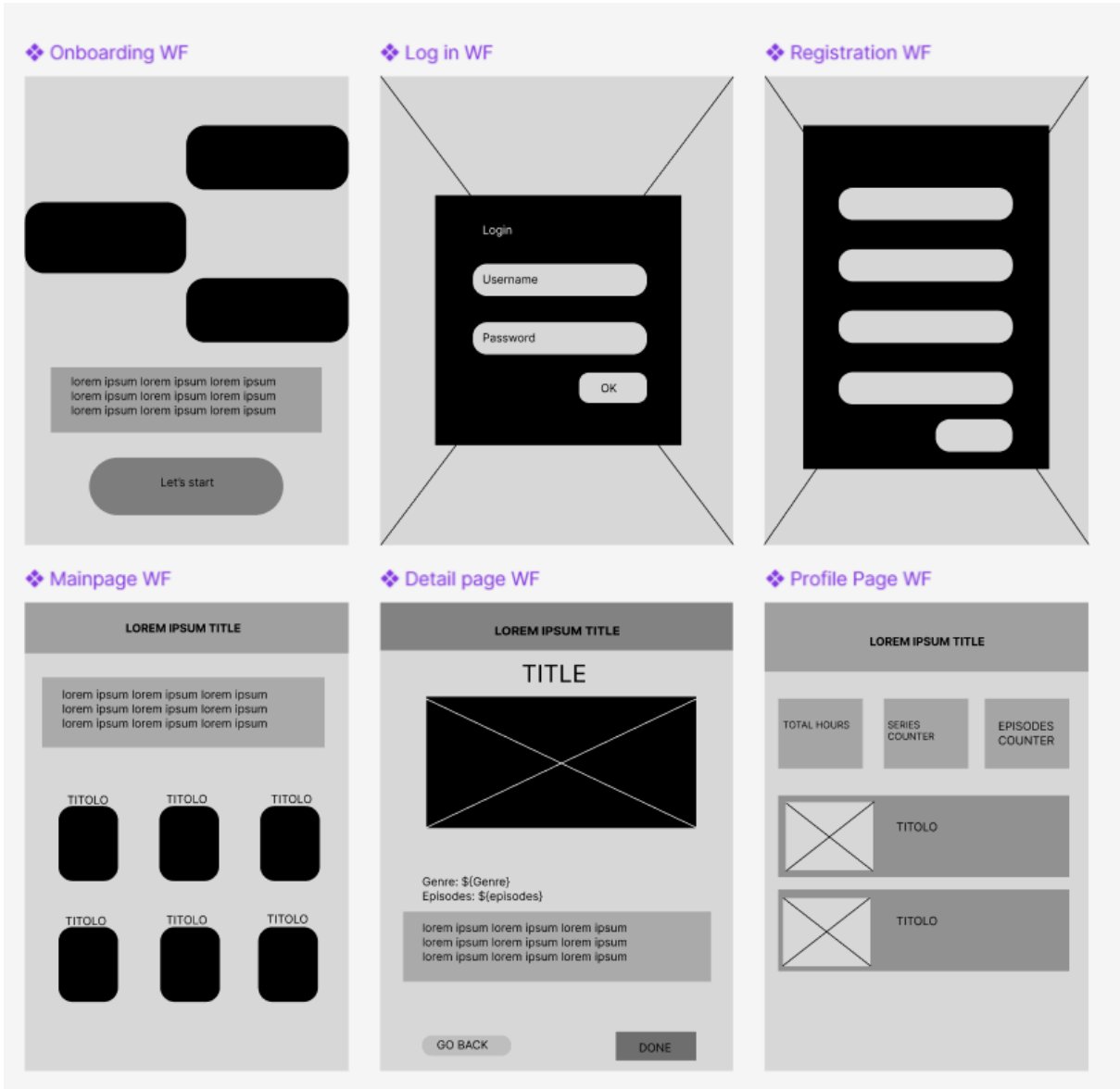
3. Stato Anime (per tracciare lo stato dell'anime per ciascun utente)

- ID_Stato (PK)
- ID_Utente (FK)
- ID_Anime (FK)
- Stato ("Completato")

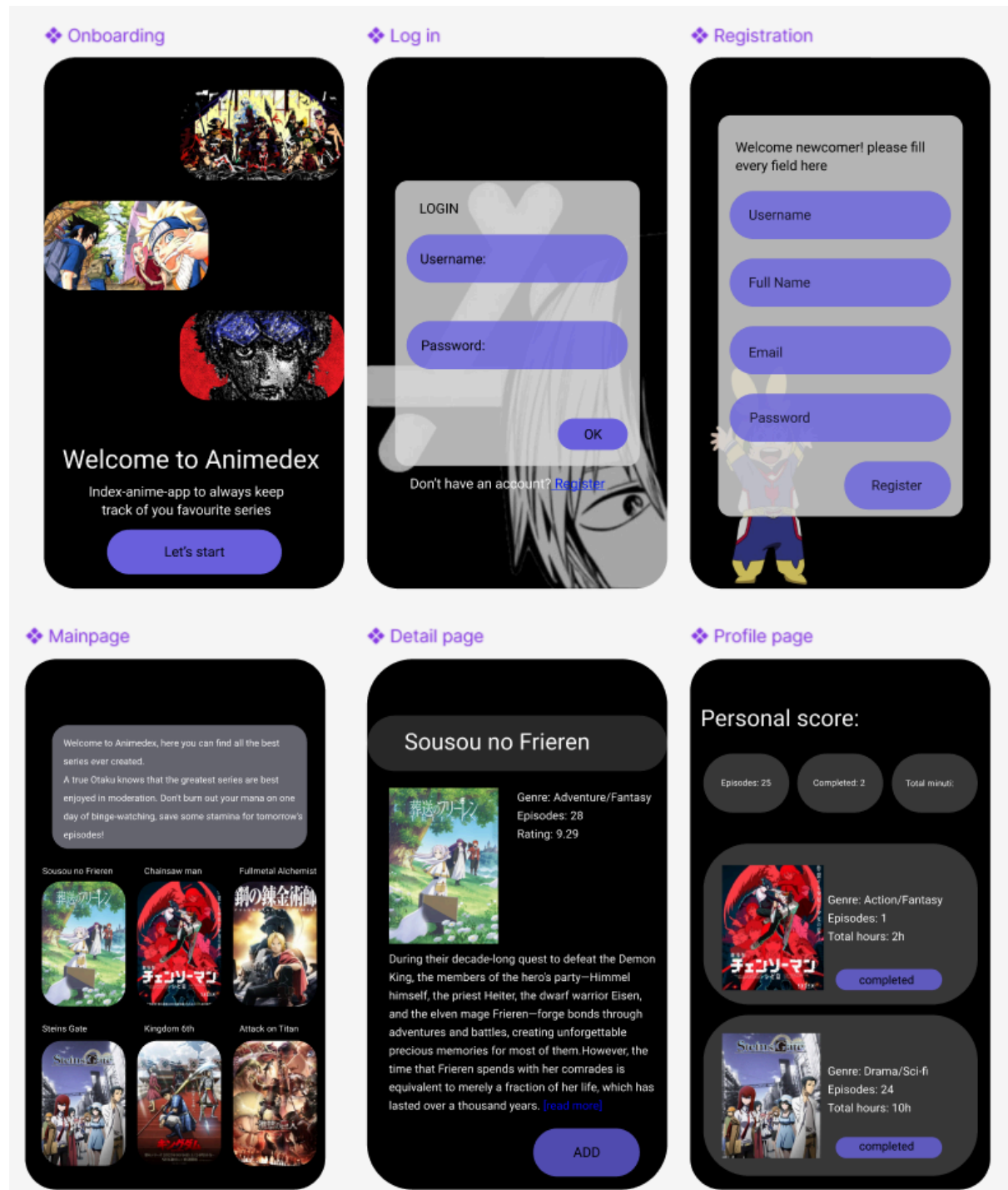
4. Dashboard Statistiche

- ID_Statistica (PK)
- ID_Utente (FK)
- Totale_Completati

Wireframe



Design



Architettura del Sistema

- **Database: SQLite** per la gestione locale dei dati, leggero e integrato nell'app Android.
- **Backend: Node.js con Express** per la normalizzazione e l'elaborazione dei dati provenienti dalle API esterne.
- **Servizi Esterni: API Jikan** per ottenere dati su anime e manga.
- **Frontend / Mobile:**
Android Studio con Java
Android 13 (Tiramisu)
- **Versionamento del codice: GitHub** per il controllo delle versioni
- **Testing: manuale** per verificare il corretto funzionamento delle chiamate API, assicurando che tutti i dati vengano recuperati correttamente e gestiti dal backend.