

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E
MATEMATICA APPLICATA



Corso di Laurea in Ingegneria Informatica

GymZone: progetto e sviluppo di un'applicazione iOS per il fitness.

Design della GUI, gestione delle notifiche e presentazione statistiche.

Relatore:
Ch. Prof.
Luca GRECO

Candidato:
Mattia DE
BARTOLOMEIS
Mat. 0612705104

ANNO ACCADEMICO 2022/2023

In memoria di mio padre, dedico questa tesi, motivato dal desiderio di rendere orgoglioso colui che mi ha insegnato il valore dell'impegno e della perseveranza.

Ringraziamenti

In primo luogo, desidero esprimere un ringraziamento speciale al mio relatore, Luca Greco, per i suoi preziosi suggerimenti e le utili indicazioni nel corso di questi mesi. La sua guida è stata fondamentale per il mio percorso accademico.

Un sentito ringraziamento va a mia madre e mia sorella per la loro amorevole presenza e il costante sostegno che mi hanno dato lungo tutto il percorso accademico. Senza di loro, non avrei potuto affrontare tutte queste sfide e sono grato per tutto ciò che hanno fatto per me.

Desidero ringraziare anche mio fratello Piersilvio, che con la sua straordinaria intelligenza è stato una fonte continua di ispirazione nel mio percorso accademico. Grazie per essere un esempio di dedizione e successo.

A mia nonna Adriana, vorrei dedicare alcune parole speciali perché è un pilastro fondamentale nella mia vita. Sei sempre stata presente e ti sono grato per tutto l'amore e il supporto che mi hai dato.

Un ringraziamento speciale va a zia Manuela e zio Luca per l'enorme affetto e il sostegno emotivo che mi hanno offerto. Nonostante la distanza geografica che ci separa, la vostra costante presenza mi fa sentire sempre vicino a voi.

Desidero esprimere la mia gratitudine alla mia famiglia paterna per la loro disponibilità e il sostegno che mi hanno offerto nel corso di questi anni.

Desidero ringraziare Davide e Francesco per essere stati sempre al mio fianco. I momenti che abbiamo trascorso insieme, sia quelli piacevoli che quelli difficili, sono stati fondamentali per raggiungere questo traguardo.

Infine, desidero ringraziare tutti i miei compagni di corso con i quali ho condiviso questi tre anni. Sono grato di aver incontrato persone così affini a me, determinate nel perseguire un obiettivo e che mi hanno costantemente spinto a dare il massimo.

Sommario

| | |
|-----------------------------------------------------------|-----------|
| Introduzione | 5 |
| 1 Panoramica delle tecnologie utilizzate | 7 |
| 1.1 Tecnologie hardware | 7 |
| 1.1.1 MacBook Pro | 7 |
| 1.1.2 Iphone 13 | 7 |
| 1.2 Tecnologie software | 8 |
| 1.2.1 XCode | 8 |
| 1.2.2 Swift | 9 |
| 1.2.3 SwiftUI | 9 |
| 1.2.4 UserNotifications | 10 |
| 1.2.5 Charts | 10 |
| 1.2.6 CoreML | 11 |
| 1.2.7 HealthKit | 11 |
| 1.2.8 SF Symbols | 11 |
| 2 Progettazione della soluzione | 13 |
| 2.0.1 Challenge Based Learning (CBL) | 13 |
| 2.0.2 Il nostro processo di CBL | 13 |
| 2.0.3 Progettazione delle interfacce utente | 15 |
| 2.0.4 Progettazione dei dati | 17 |
| 3 Implementazione e descrizione delle funzionalità | 20 |
| 3.0.1 WorkoutsView | 20 |
| 3.0.2 AddWorkoutView | 22 |
| 3.0.3 AddExerciseView | 23 |
| 3.0.4 ExercisePropertyView | 24 |
| 3.0.5 WorkoutView | 25 |
| 3.0.6 PlayView | 25 |
| 3.0.7 AwardView | 28 |
| 3.0.8 ExecutionView | 29 |
| 3.0.9 SingleExecutionView | 30 |
| 3.0.10 ProfileView | 31 |
| 3.0.11 ChartView | 32 |
| 3.0.12 AnalyticsView | 33 |
| 3.0.13 PredictionView | 33 |

| | |
|-------------------------------------------|-----------|
| 3.0.14 Gestione delle notifiche | 34 |
| 4 Conclusioni e prospettive future | 36 |

Introduzione

Nell'era digitale in cui ci ritroviamo, il fitness e il benessere personale stanno assumendo sempre più rilevanza. Infatti, sono sempre di più le persone che provano ad adottare uno stile di vita sano e attivo, cercando di raggiungere i propri obiettivi allenandosi. Su questo tema, le applicazioni riguardanti il mondo del fitness si sono affermate come elementi fondamentali per aiutare le persone a mantenersi in forma adottando uno stile di vita salutare. Grazie a queste applicazioni, gli utenti possono monitorare e registrare i propri progressi, accedere a programmi di allenamento personalizzati, seguire piani alimentari specifici e partecipare a sfide o competizioni con altri utenti. Tutto ciò rende il fitness un'esperienza interattiva e motivante, consentendo agli utenti di raggiungere i propri obiettivi in modo più divertente.

La tesi che verrà presentata nasce dall'esperienza ottenuta durante il tirocinio svolto seguendo l'Apple Foundation Program, in cui ho avuto l'opportunità di progettare e implementare un'applicazione iOS innovativa con un team composto da due persone, ognuna delle quali ha contribuito allo sviluppo principale dell'applicazione, concentrandosi successivamente in maniera individuale su funzionalità specifiche. Durante il tirocinio, abbiamo adottato la "Challenge Based Learning", un metodo di apprendimento fortemente sostenuto da Apple. Questo approccio coinvolge attivamente gli studenti nell'identificazione, nell'analisi e nella progettazione di soluzioni per problemi reali. Partendo da un'idea fondamentale, ci siamo impegnati a sviluppare una soluzione finale concreta. L'idea principale da cui noi siamo partiti è il Fitness e sono orgoglioso di presentare la nostra soluzione: **GymZone**.

GymZone è un'applicazione per l'allenamento in palestra. Il suo obiettivo principale è offrire agli utenti uno strumento completo ed efficace per monitorare e migliorare il proprio percorso di allenamento. All'interno dell'applicazione l'utente può creare ed eseguire allenamenti personalizzati, tenendo traccia di tutti i pesi utilizzati durante le sessioni di allenamento. Inoltre, è presente una sezione profilo dedicata, che consente all'utente di salvare i propri dati personali per tener traccia dei progressi futuri. In questa sezione l'utente può inserire informazioni come il peso corporeo, le circonferenze dei vari gruppi muscolari e altri dati rilevanti. Inoltre, l'applicazione GymZone offre anche una sezione dedicata alle esecuzioni dei singoli esercizi. Qui, gli utenti possono visualizzare gif dimostrative e descrizioni dettagliate per ogni esercizio, fornendo una guida visiva e testuale per eseguire correttamente gli allenamenti. Infine, l'utente ha la possibilità di ottenere una stima delle calorie che brucerà in futuro, attraverso una sezione dedicata dell'applicazione, Analytics, in cui vengono richieste alcune informazioni specifiche per effettuare la stima.

Questa tesi si propone di presentare il progetto GymZone e di illustrare il mio contributo personale nello sviluppo di questa applicazione. Oltre alla partecipazione alla realizzazione dei componenti principali dell'applicazione, ho avuto l'opportunità di occuparmi individualmente della gestione delle notifiche locali, dell'implementazione della sezione "Profilo", con un grafico che mostra l'andamento del peso corporeo nel tempo e di altre funzionalità minori, che verranno descritte in seguito.

Nel capitolo 1, fornirò una panoramica sulle tecnologie hardware e software utilizzate per l'implementazione dell'applicazione. In particolare saranno approfonditi il principale linguaggio di programmazione e i framework utilizzati nel processo di sviluppo dell'applicazione.

Nel capitolo 2, mi concentrerò sulla fase progettuale dell'applicazione GymZone, in particolare andando ad analizzare tutte le fasi del Challenge Based Learning che ci hanno guidato verso la soluzione finale. Saranno descritti gli obiettivi che ci siamo prefissati, le sfide che abbiamo affrontato e le strategie adottate per risolverle.

Nel capitolo 3, presenterò i risultati dell'implementazione di GymZone, spiegando nel dettaglio tutte le funzionalità dell'applicazione e includendo immagini a supporto.

Infine, nel capitolo 4, trarrò le conclusioni principali dal mio lavoro e discuterò delle possibili direzioni future per il miglioramento di GymZone e per ulteriori ricerche nel campo delle applicazioni fitness. Saranno valutati i punti di forza e le limitazioni dell'applicazione, insieme a suggerimenti per ampliare le sue funzionalità e migliorare l'esperienza dell'utente.

Attraverso questo elaborato, proveremo ad evidenziare il valore e l'impatto che l'applicazione GymZone può avere nel promuovere uno stile di vita sano e attivo, fornendo agli utenti uno strumento personalizzato per il raggiungimento dei loro obiettivi di fitness.

Capitolo 1

Panoramica delle tecnologie utilizzate

In questo capitolo esploreremo le tecnologie utilizzate per lo sviluppo dell'applicazione GymZone. Concentreremo l'attenzione sulle tecnologie hardware fornite dall'università stessa, quali MacBook Pro e iPhone 13 e sulle tecnologie software fra cui il linguaggio di programmazione utilizzato, l'ambiente di sviluppo e altri framework utili per lo sviluppo dell'applicazione.

1.1 Tecnologie hardware

Nella prima sezione esamineremo le tecnologie hardware utilizzate per lo sviluppo dell'applicazione:

1. MacBook Pro
2. iPhone 13

1.1.1 MacBook Pro

Uno degli elementi chiave forniti dall'università per lo sviluppo dell'applicazione è stato il MacBook Pro M1 [1]. Il MacBook Pro M1 rappresenta una significativa evoluzione nell'architettura dei processori di Apple, offrendo prestazioni elevate e un'efficienza energetica superiore. Questo dispositivo ci ha consentito di lavorare in modo efficiente ed efficace durante tutto il ciclo di sviluppo dell'applicazione. La potenza di elaborazione del processore M1 e l'integrazione ottimizzata tra hardware e software hanno reso possibile l'esecuzione di complesse operazioni di sviluppo e test.

1.1.2 Iphone 13

Un'altro strumento fondamentale per lo sviluppo dell'applicazione è stato l'iPhone 13 [2] attraverso il quale è stato possibile effettuare il testing dell'applicazione direttamente su un dispositivo fisico.

L'iPhone 13 è dotato di funzionalità avanzate e prestazioni elevate. Utilizzando un dispositivo reale per il testing, siamo stati in grado di valutare le prestazioni e l'usabilità dell'app su un dispositivo mobile.

1.2 Tecnologie software

In questa sezione esamineremo le tecnologie software utilizzate per lo sviluppo dell'applicazione:

1. XCode
2. Swift
3. SwiftUI
4. UserNotifications
5. Charts
6. CoreML
7. HealthKit
8. SF Symbols

Per lo sviluppo completo dell'applicazione sono stati impiegati anche i framework CoreML e HealthKit, che descriveremo brevemente poiché non sono stati oggetto del mio contributo personale.

1.2.1 XCode

Xcode [3] è stato l'ambiente di sviluppo integrato (IDE) utilizzato per sviluppare l'applicazione GymZone. Fornito da Apple, Xcode rappresenta uno strumento essenziale per gli sviluppatori iOS, offrendo una vasta gamma di funzionalità che semplificano il processo di sviluppo. Di seguito elencheremo alcuni vantaggi dell'utilizzo di tale IDE. Attraverso la sua interfaccia intuitiva, Xcode mette a disposizione un insieme completo di strumenti necessari per la scrittura del codice, il debugging, la gestione delle risorse e la creazione dell'interfaccia utente. Il suo editor di codice offre funzionalità avanzate, come l'autocompletamento, il controllo della sintassi e la segnalazione degli errori, facilitando la scrittura di codice pulito e corretto.

Inoltre, Xcode dispone di un potente compilatore che traduce il codice scritto in Swift in linguaggio macchina comprensibile per i dispositivi iOS. Questo processo di compilazione consente di generare il codice eseguibile dell'applicazione pronto per essere installato e testato su un dispositivo fisico.

Un'altra caratteristica rilevante di Xcode è l'Interface Builder, uno strumento visuale integrato che consente di progettare l'interfaccia utente dell'applicazione in modo intuitivo e interattivo. Utilizzando l'Interface Builder, è possibile posizionare e configurare gli elementi dell'interfaccia, definire le relazioni tra di essi e impostare i vincoli di layout per garantire una corretta visualizzazione su dispositivi di diverse dimensioni. Tale funzionalità non è stata da noi sfruttata, avendo utilizzato come framework SwiftUI e non UIKit. Infatti attraverso SwiftUI l'interfaccia grafica è creata direttamente tramite Swift a differenza di UIKit che permette di progettare l'interfaccia direttamente in modo visuale e gestirne il funzionamento logico tramite Swift.

Infine, Xcode offre anche strumenti per il testing e il debug dell'applicazione, consentendo di individuare e risolvere eventuali errori o problemi di performance. Con il simulatore iOS integrato, è possibile eseguire e testare l'applicazione su una varietà di modelli di iPhone e iPad senza la necessità di un dispositivo fisico. In alternativa è

possibile testare l'applicazione anche direttamente sul proprio dispositivo apple , collegandolo al MacBook su cui si sta sviluppando l'applicazione. Nel nostro caso sono stati effettuati test parziali sfruttando il simulatore iOS integrato e test completi utilizzando l'iPhone 13 fornitoci dall'università.

1.2.2 Swift

Swift [4] è stato il linguaggio di programmazione utilizzato per sviluppare l'applicazione GymZone. Introdotta da Apple, Swift rappresenta una scelta moderna, sicura e potente per lo sviluppo di applicazioni iOS.

L'aspetto principale di tale scelta risiede nel fatto che Swift è integrato strettamente con l'ecosistema Apple, fornendo accesso completo alle librerie e alle API di sistema. Questo ha facilitato l'interazione con le funzionalità specifiche di iOS e ha consentito di sfruttare al meglio le potenzialità della piattaforma. In particolare, sfruttando tali librerie, è stato possibile gestire le notifiche e ottenere dati direttamente da Health per implementare funzionalità che vedremo in seguito.

Inoltre, Swift offre una vasta gamma di funzionalità che semplificano lo sviluppo delle applicazioni. Tra queste ci sono le opzioni di tipizzazione forte, che aumentano la sicurezza del codice e consentono la rilevazione anticipata degli errori durante la compilazione. Infatti per dichiarare una variabile o una costante , è necessario specificarne il tipo che non sarà più modificabile in seguito. Questo è un aspetto che differenzia Swift da altri linguaggi di programmazione come Python, in cui è utilizzato un approccio di tipizzazione dinamica. Inoltre, Swift supporta il controllo dei flussi avanzato attraverso i costrutti come if, switch e while. Permette infine l'uso di optional per gestire i valori nulli in modo sicuro. In particolare un optional è una variabile che può assumere un valore o essere nil e si dichiara inserendo un '?' dopo la dichiarazione del tipo della variabile.

Un altro aspetto significativo di Swift è la sua performance. Grazie alla sua progettazione ottimizzata, Swift offre prestazioni elevate, consentendo un'esperienza fluida per gli utenti dell'applicazione.

1.2.3 SwiftUI

SwiftUI [5] è il framework fornito da Apple, utilizzato per la realizzazione dell'interfaccia utente per l'applicazione GymZone. La scelta di SwiftUI e non dell'altro framework UIKit messo a disposizione sempre da Apple, risiede nei seguenti motivi:

SwiftUI utilizza un approccio dichiarativo per la creazione dell'interfaccia utente, consentendo di descrivere la struttura e l'aspetto dell'interfaccia attraverso una sintassi chiara e semplice. Con SwiftUI, è possibile dichiarare quali componenti grafici si desidera utilizzare, come dovrebbero essere posizionati e come dovrebbero interagire tra loro. Tale approccio semplifica la struttura del codice riducendone la complessità.

D'altra parte UIKit utilizza un approccio imperativo per la creazione dell'interfaccia utente. Ciò comporta la scrittura di codice dettagliato per creare e gestire i componenti dell'interfaccia utente.

Un altro vantaggio importante di SwiftUI è la velocità di sviluppo. E' possibile infatti avere una live preview del nostro codice, senza dover testare sempre l'applicazione sul simulatore.

L'unico svantaggio di tale scelta è che SwiftUI richiede iOS 13 come versione minima del sistema operativo, ma ciò è stato ritenuto poco rilevante ai fini del nostro progetto.

1.2.4 UserNotifications

UserNotifications [6] è un framework fornito da Apple per gestire le notifiche all'interno delle applicazioni iOS. È molto utilizzato per inviare notifiche locali agli utenti, per poter informarli, coinvolgerli o ricordare loro determinati eventi all'interno dell'applicazione. Come illustreremo più dettagliatamente nei prossimi capitoli, tale funzionalità mantiene un ruolo importante nella nostra applicazione. Infatti permette di ricordare agli utenti l'orario impostato per la seduta di allenamento, stimolandoli a non rimandare il loro workout. Su quest'argomento è bene aprire una piccola parentesi per differenziare quelle che sono le notifiche locali, utilizzate nella nostra applicazione, dalle notifiche remote.

Notifiche locali Le notifiche locali sono gestite e generate direttamente all'interno dell'applicazione sul dispositivo dell'utente senza richiedere una connessione di rete. Queste sono infatti programmate in base a data specifica o un'area geografica in cui si trova l'utente.

Notifiche remote Le notifiche push sono inviate ad un dispositivo tramite una connessione di rete. Sono generate da un server remoto ed inviate al dispositivo. Esse possono essere utilizzate allo scatenarsi di eventi. Un esempio pratico di utilizzo lo si ha nelle applicazioni di messaggistica, in cui l'utente riceve una notifica push quando riceve un nuovo messaggio.

1.2.5 Charts

Charts [7] è un framework fornita da Apple per la realizzazione di grafici all'interno delle applicazioni iOS. E' utilizzato per semplificare la visualizzazione dei dati, rendendoli più intuitivi e per migliorare la User Experience dell'utente. Come illustreremo più dettagliatamente nei prossimi capitoli, l'utilizzo di tale framework ci ha permesso di rappresentare dati relativi ai progressi fisici ottenuti dall'utente in maniera chiara e comprensibile. In particolare vedremo che sono stati sviluppati dei grafici che mostrano l'andamento del peso dell'utente nel tempo e predicono le calorie bruciate dall'utente, confrontandole con quelle bruciate nei giorni precedenti, in base alla sua attività fisica. Tale libreria permette di generare diversi tipi di grafici, tra cui:

- 'LineChart': E' un grafico che permette di visualizzare i dati come una serie di punti collegati da una linea.
- 'BarChart': E' un grafico che rappresenta i dati come barre verticali o orizzontali.
- 'PieChart': E' un grafico che visualizza i dati come una torta divisa in fette, dove l'area di ogni fetta rappresenta la proporzione dei dati.

Infine sono presenti opzioni di personalizzazione avanzata del grafico che permettono di modificare gli assi, le etichette, i colori, lo stile delle linee, lo sfondo e molti altri aspetti.

1.2.6 CoreML

CoreML [8] è un framework fornito da Apple per integrare all'interno della propria applicazione modelli di machine learning. Come già accennato, GymZone possiede una sezione 'Analytics' in cui inserendo dei dati è possibile visualizzare un grafico che confronta le calorie bruciate con le calorie che si bruceranno nei prossimi tre giorni. CoreML è stato quindi utilizzato per creare un modello di predizione delle calorie che a partire da minuti sedentari, minuti di attività e distanza totale percorsa dall'utente può stimare il dispendio calorico futuro. In questo modo l'applicazione è in grado di effettuare previsioni sul dispendio calorico, senza necessità di connessione internet. Il modello creato è un "Boosted Tree". I Boosted Tree combinano numerosi alberi decisionali deboli per creare una previsione più potente. Ogni albero successivo cerca di correggere gli errori commessi dagli alberi precedenti. Questo processo iterativo porta a un risultato finale più accurato nel predire l'output desiderato. I boosted tree presentano diversi vantaggi:

1. Potenza predittiva: I boosted tree sono noti per la loro potenza predittiva e sono in grado di generare modelli molto accurati.
2. Robustezza ai dati rumorosi: I boosted tree sono in grado di gestire dati rumorosi o contenenti valori mancanti senza doverli pre-elaborare in modo significativo. Gli alberi potenziati sono in grado di adattarsi a queste imperfezioni durante il processo di addestramento.
3. Interpretabilità: A differenza di alcuni altri modelli di apprendimento automatico, come le reti neurali profonde, i boosted tree sono relativamente facili da interpretare. È possibile ottenere informazioni sui fattori di decisione utilizzati dagli alberi per arrivare a una previsione.

1.2.7 HealthKit

HealthKit [9] è un framework fornito da Apple che consente agli sviluppatori di creare applicazioni che possono accedere e condividere dati relativi alla salute degli utenti. Questi dati sono letti direttamente dall'applicazione Health presente sul dispositivo iOS e includono informazioni come battito cardiaco, passi percorsi, livelli di attività fisica, sonno, peso corporeo e molto altro. Nel contesto della mia applicazione, tale framework è stato utilizzato per acquisire i dati relativi alle calorie bruciate degli utenti. Infatti questi dati sono stati utilizzati insieme a quelli prodotti dal modello di predizione delle calorie creato con CoreML per creare un grafico di confronto. Affinchè ciò sia realizzabile, è necessario che gli utenti forniscano il consenso per condividere i loro dati sanitari con l'applicazione. Il consenso può essere poi revocato in qualsiasi momento se desiderano interrompere la condivisione dei loro dati con l'applicazione.

1.2.8 SF Symbols

SFSymbols è una libreria di icone di alta qualità sviluppata da Apple e progettata appositamente per adattarsi ai dispositivi iOS. Tale libreria offre una vasta gamma di simboli, che coprono diverse categorie d'interesse come sport, trasporti, messaggistica e molto altro. Abbiamo integrato la libreria di icone SF Symbols nella nostra applicazione, concentrandoci principalmente sulla ricerca di icone adatte per le pagine principali della nostra applicazione. Fra le icone disponibili abbiamo scelto le seguenti:

- "dumbbell.fill", indica un manubrio di pesi ed è stata scelta per rappresentare la sezione "Workouts".
- "figure.strengthtraining.traditional", indica una figura che solleva pesi ed è stata scelta per rappresentare la sezione "Executions".
- "chart.bar.xaxis", indica un grafico a barre orizzontali ed è stata scelta per rappresentare la sezione "Analytics".
- "person.fill", indica una silhouette di una persona ed è stata scelta per rappresentare la sezione "Profile".

Capitolo 2

Progettazione della soluzione

In questo capitolo esploreremo la fase di identificazione, analisi e progettazione della nostra applicazione. In particolare, ci immergeremo in un metodo chiamato 'Challenge Based Learning' (CBL), che verrà approfondito in seguito per consentire una migliore comprensione del resto della tesi. Una volta ottenuta la soluzione tramite il CBL, ci dedicheremo alla fase successiva, ossia la progettazione delle interfacce utente che prenderanno vita e la gestione dei dati.

2.0.1 Challenge Based Learning (CBL)

Il Challenge Based Learning [10] è un approccio educativo che si basa sulla risoluzione di sfide e problemi complessi. Tale approccio coinvolge gli studenti in maniera attiva, incoraggiandoli a trovare soluzioni innovative per la sfida preposta. E' possibile dividere il CBL in tre fasi:

1. Fase di coinvolgimento: in questa fase si identifica una 'Big Idea' ovvero una tematica generale su cui incentrare la sfida e una serie di 'Essential Questions', ovvero ci si pone delle domande molto generali relative alla Big Idea che permetteranno poi di identificare la sfida che si vuole intraprendere.
2. Fase di indagine: l'obiettivo della fase di indagine è condurre ricerche rigorose, basate su contenuti e concetti per creare una base per soluzioni attuabili e sostenibili. Nella pratica in tale fase si formulano delle 'Guiding Questions', ovvero delle domande mirate orientate verso la sfida individuata nella fase di coinvolgimento. A partire da queste domande vi è una fase di ricerca in cui si trovano delle risposte e si traggono delle conclusioni.
3. Fase di azione: In questa fase, basandosi sulle ricerche effettuate si arriva alla soluzione ideale per la nostra sfida.

2.0.2 Il nostro processo di CBL

In questo paragrafo vedremo come il processo di CBL è stato utilizzato ai fini della progettazione di GymZone. In particolare esploreremo nel dettaglio ogni fase di tale approccio che ci ha condotti alla soluzione finale.

Fase di coinvolgimento Essendo in prima persona un grande appassionato del mondo della palestra, la scelta riguardante la 'BIG IDEA' è stata abbastanza semplice: il Fitness. Relativamente a quest'argomento sono state stilate le seguenti 'Essential Questions':

- In che modo l'essere seguiti da un allenatore migliora l'allenamento?
- Qual è l'utilità di un allenatore nel proprio sport?
- Quali sono i principali fattori che influenzano il fitness?
- Come possono i principianti non allenati, senza un personal trainer, assicurarsi che le loro esecuzioni siano corrette?
- Quali sono i modi per misurare il progresso nel fitness?
- Quali sono i fattori che influenzano la motivazione per allenarsi?
- Come poter allenarsi da soli senza andare in palestra?

Attraverso queste domande, abbiamo capito quale sarebbe stata la sfida da affrontare. **Challenge:** "Migliorare l'allenamento attraverso un'organizzazione efficiente, tracciando gli allenamenti e fornendo metodi di esecuzione corretti per qualsiasi tipo di esercizio".

Ricapitolando, la fase di "coinvolgimento" del progetto CBL ha coinvolto l'identificazione della "BIG IDEA" legata al fitness, la formulazione di domande essenziali per esplorare gli aspetti chiave del fitness e la definizione di una challenge finale concentrata sull'organizzazione degli allenamenti, il monitoraggio dei risultati e i metodi di esecuzione corretti.

Fase di indagine Durante la fase di indagine abbiamo pensato e posto delle 'Guiding Questions' a diverse persone attraverso un modulo Google. Di seguito sono riportate le domande con i risultati ottenuti e le conclusioni a cui siamo giunti:

- Quanto ritieni importante il fitness nella vita quotidiana? Abbiamo scoperto che la maggior parte delle persone ritiene il fitness molto importante per la vita quotidiana.
- Quanto ritieni importante seguire una dieta equilibrata nella vita quotidiana? La maggior parte delle persone ritiene importante seguire una dieta equilibrata nella vita quotidiana.
- Quanto può essere conveniente avere un piano alimentare e di allenamento unificato e registrato in un unico luogo facilmente accessibile? Molte persone hanno risposto che sarebbe molto conveniente avere un piano alimentare e di allenamento unificato e registrato in un unico posto facilmente accessibile.
- Può essere utile avere un diario per registrare gli allenamenti in palestra al fine di avere un confronto con quelli precedenti? La maggior parte delle persone ha risposto che sarebbe utile avere un diario per registrare gli allenamenti in palestra e confrontarli con quelli precedenti.
- Ti è mai capitato di dimenticare l'esecuzione corretta di un esercizio? Molte persone hanno risposto di aver dimenticato l'esecuzione corretta di un esercizio.

- Può essere utile avere un sistema automatico che conta serie, ripetizioni o tempi di recupero? La maggior parte delle persone ha risposto che sarebbe utile avere un sistema automatico che conta serie, ripetizioni o tempi di recupero.
- Può essere utile avere un modo per ricevere un piano di allenamento dal proprio allenatore e fornirgli i propri dati per studiare nuovi piani di allenamento? La maggior parte delle persone ha risposto che sarebbe utile avere un modo per ricevere un piano di allenamento dal proprio allenatore e condividere i propri dati con lui.
- Quanto ritieni importante avere un piano di allenamento settimanale per ottenere buoni risultati? La maggior parte delle persone ha risposto che è molto importante avere un piano di allenamento settimanale per ottenere buoni risultati.
- Può essere utile avere un sistema che guidi l'esecuzione corretta dell'esercizio al fine di mantenere la postura corretta? Molte persone hanno risposto che sarebbe utile avere un sistema che guidi l'esecuzione corretta dell'esercizio per mantenere la postura corretta.
- Hai bisogno di essere motivato per allenarti? La maggior parte delle persone ha risposto di avere bisogno di essere motivato per allenarsi.

A partire da quest'analisi siamo giunti alla conclusione che le persone hanno bisogno di essere più organizzate durante i loro allenamenti.

Fase di azione Sulla base delle ricerche effettuate siamo giunti alla nostra soluzione: **Soluzione:** Un'applicazione su cui organizzare e monitorare l'allenamento delle persone permettendo di tracciare il peso e le ripetizioni, evidenziando i progressi ottenuti e fornendo istruzioni precise per l'esecuzione corretta di ogni esercizio.

2.0.3 Progettazione delle interfacce utente

Le interfacce giocano un ruolo fondamentale nell'esperienza dell'utente con le applicazioni digitali. In questa sezione analizzeremo quest'aspetto da un punto di vista generale per poi focalizzarci sulla nostra applicazione.

Un'interfaccia utente deve essere progettata garantendo i seguenti elementi [11]:

- Integrità estetica: l'aspetto e il comportamento dell'app devono integrarsi in modo armonioso con la sua funzione. Ciò significa, ad esempio, che l'interfaccia di un gioco non potrà essere uguale a quella utilizzata per un'applicazione di fitness.
- Coerenza: l'interfaccia dovrebbe seguire standard e paradigmi familiari, utilizzando elementi dell'interfaccia forniti dal sistema, icone conosciute, stili di testo standard e terminologia uniforme.
- Manipolazione diretta: l'interfaccia dovrebbe consentire agli utenti di manipolare direttamente i contenuti sullo schermo, attraverso gesti o movimenti del dispositivo.
- Feedback: l'interfaccia dovrebbe fornire dei feedback percepibili in risposta alle azioni dell'utente, ad esempio attraverso evidenziazione degli elementi interattivi quando vengono toccati o indicatori di progresso.

- Conversazione con gli utenti: ogni parola nell'app fa parte di una conversazione con gli utenti. E' bene quindi utilizzare parole e frasi familiari e comprensibili, evitando termini tecnici che potrebbero non essere compresi.
- Terminologia: bisognerebbe utilizzare un tono informale e amichevole, facendo attenzione all'uso dell'umorismo, poiché ciò che potrebbe sembrare divertente inizialmente potrebbe diventare irritante nel tempo. Inoltre, l'umorismo può non essere traducibile in altre culture.
- Tipografia: Apple consiglia di utilizzare il carattere sans-serif "San Francisco" (SF) come font di sistema in iOS, macOS, tvOS e il carattere "New York" (NY) in contesti grafici e di lettura (ad esempio per titoli o testi di dimensione maggiore).

Queste sono le linee guida fornite da Apple per la progettazione dell'interfaccia utente. Un altro aspetto importante da tenere a mente è l'User Experience(UX) [12], ovvero le percezioni e le risposte degli utenti che derivano dall'uso dell'applicazione. Per garantire una buona user experience, diversi requisiti devono essere soddisfatti:

- Innanzitutto, l'interfaccia deve essere facile da usare e comprensibile per gli utenti. Ciò implica che gli elementi grafici devono essere chiaramente visibili, con un buon feedback che fornisce informazioni sulle azioni dell'utente e i risultati ottenuti. Inoltre, l'interfaccia dovrebbe essere intuitiva, consentendo agli utenti di apprendere facilmente come utilizzare il sistema senza dover consultare istruzioni complesse o fare sforzi eccessivi.
- Un'altra componente importante per una buona UX è l'efficienza. L'interfaccia dovrebbe permettere agli utenti di raggiungere i loro obiettivi in modo rapido. Ciò implica che le operazioni più comuni siano semplificate e che siano presenti scorciatoie per arrivare all'obiettivo finale.
- Infine, la UX richiede anche che l'interfaccia dovrebbe essere progettata tenendo conto delle caratteristiche e delle esigenze degli utenti specifici, adattandosi alle loro capacità e conoscenze. Questo potrebbe significare fornire opzioni di personalizzazione come il supporto per diverse lingue.

Le nostre interfacce utente sono state progettate seguendo tutte le linee guida fornite da Apple. In particolare esse garantiscono: integrità estetica, coerenza, manipolazione diretta, feedback. Inoltre la progettazione ha tenuto conto anche dell'aspetto legato alla User Experience, cercando di semplificare il più possibile le operazioni, evitando una registrazione iniziale per accedere all'app e disponendo gli elementi grafici in maniera chiara e visibile.

GymZone, per poter soddisfare tutte le funzionalità richieste, si suddivide in 4 sezioni principali:

1. Workouts: La sezione 'Workouts' rappresenta la 'home' dell'applicazione e contiene tutti gli allenamenti dell'utente.
2. Executions: La sezione 'Executions' contiene tutti gli esercizi disponibili nell'applicazione e consente di accedere a una pagina dettagliata che fornisce una descrizione e una dimostrazione completa di come svolgere l'esercizio.
3. Analytics : La sezione 'Analytics' contiene una serie di campi di input che serviranno per stimare le calorie bruciate dall'utente in futuro.

4. Profile: la sezione 'Profile' che raccoglie tutti i dati dell'utente.

Come dimostrazione delle considerazioni precedenti, ecco un'anteprima della schermata Workouts:

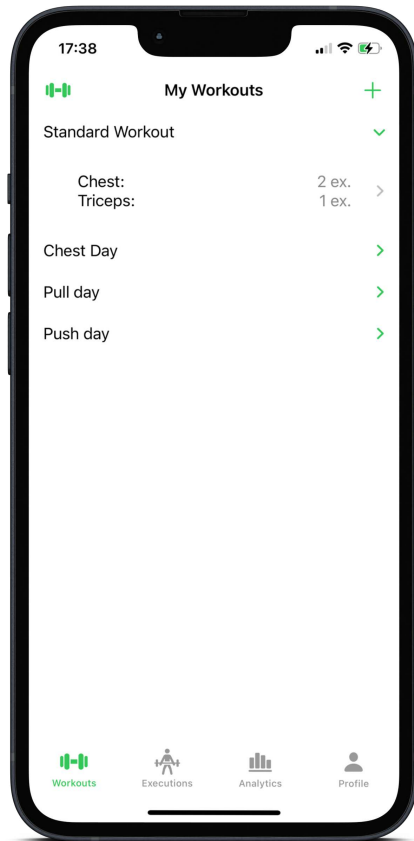


Figura 2.1: Schermo chiaro

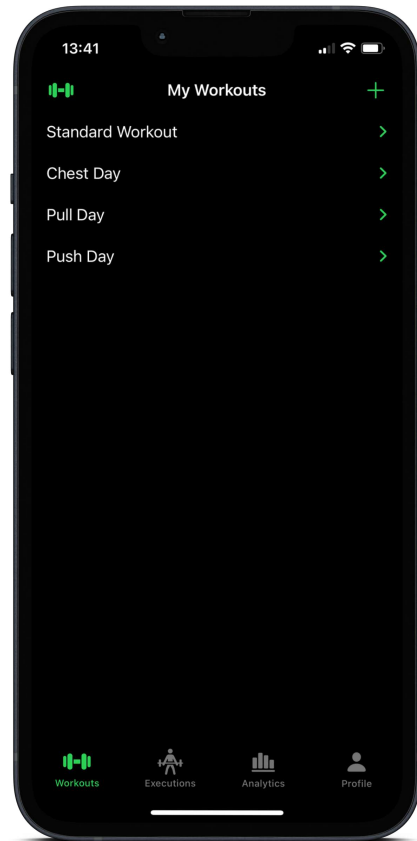


Figura 2.2: Schermo scuro

Come possiamo notare dalle figure, l'integrità estetica è rispettata, infatti sono stati utilizzati elementi grafici sobri e semplici essendo un applicazione di fitness. Inoltre l'interfaccia risulta coerente, utilizzando solo gli elementi grafici forniti dal sistema. In particolare abbiamo scelto il colore verde per l'interfaccia al fine di migliorare la coerenza dell'app in quanto riflette il design dell'app per il fitness sul watch. L'interfaccia possiede anche manipolazione diretta, anche se tale caratteristica non può essere notata dagli screenshot forniti. Presenta dei feedback visivi al tocco dei diversi elementi e al completamento dei workout eseguiti. E' infine rispettata la tipografia e utilizza una terminologia adatta, evitando umorismo e termini tecnici. Ulteriori immagini sulle interfacce utente dell'applicazione saranno fornite direttamente nel capitolo successivo, in cui verrà eseguita anche un'analisi dettagliata sulla fase di implementazione.

2.0.4 Progettazione dei dati

Come per la maggior parte delle applicazioni mobili, anche per il nostro progetto è richiesta una permanenza dei dati. Infatti nella nostra applicazione, l'utente deve poter memorizzare tutti gli allenamenti con i relativi pesi utilizzati per ogni esercizio e non solo.

Esistono diversi modi per garantire la permanenza dei dati, tra cui:

- Utilizzo di un database
- Salvataggio dei dati su file

Nella maggior parte dei casi è preferibile l'utilizzo di un database gestito da un DBMS apposito perché riduce le ridondanze e permette di inserire vincoli sui dati. Tuttavia nella nostra applicazione, dovendo trattare nella maggior parte dei casi dati semplici, e avendo dato priorità alla parte di sviluppo dell'applicazione, si è deciso di utilizzare 'User Defaults'. Di seguito esploreremo tutte e due le soluzioni nel dettaglio.

Utilizzo di database gestito da un DBMS L'utilizzo di un database gestito da un Database Management System (DBMS) offre numerosi vantaggi:

- In primo luogo, riduce le ridondanze dei dati, consentendo di archiviare le informazioni in modo efficiente ed evitando la duplicazione inutile.
- In secondo luogo, un DBMS permette di impostare vincoli sui dati, come chiavi primarie, chiavi esterne e vincoli di integrità, che garantiscono l'accuratezza e la coerenza dei dati memorizzati. Questo assicura che tutti i dati memorizzati siano corretti e coerenti.

L'approccio corretto per l'utilizzo di un database si basa su una fase di progettazione e una fase di implementazione.

La progettazione del database si suddivide generalmente in tre fasi:

- La progettazione concettuale è la fase iniziale della progettazione del database ed è incentrata sulla modellazione concettuale dei dati. Durante questa fase, vengono identificate le entità principali coinvolte nel sistema e le relazioni tra di esse. In questa fase si genera un modello Entità-Associazione per rappresentare le entità, gli attributi e le associazioni tra di esse.
- La progettazione logica è la fase successiva in cui il modello concettuale viene trasformato in un modello logico, più vicino al modo in cui i dati saranno effettivamente memorizzati nel database. Ad esempio, se stiamo andando verso una base di dati relazionale, vengono eliminati attributi composti e multivalore e eventuali generalizzazioni. Infine le entità, gli attributi e le relazioni definite nel modello concettuale vengono mappate in tabelle, colonne e vincoli di un database relazionale.
- La progettazione fisica è la fase finale della progettazione del database ed è incentrata sulla definizione delle caratteristiche di implementazione fisica del database.

Successivamente nella fase di implementazione viene creato il database attraverso il DBMS selezionato, vengono create le tabelle di riferimento, caricati i dati e implementati i vincoli sui dati. Una volta realizzato il database, è possibile interfacciarsi dall'applicazione mediante l'uso di framework come 'CoreData' nel caso di Swift.

User Defaults UserDefaults è un meccanismo di gestione dei dati in iOS che consente di salvare e recuperare dati di piccole dimensioni all'interno dell'applicazione. In termini semplici, UserDefaults può essere considerato come un piccolo "archivio" all'interno dell'applicazione dove è possibile memorizzare dati come stringhe, numeri, booleani e oggetti. Questi dati vengono salvati su un file di preferenze dell'applicazione che viene gestito dal sistema operativo. Una delle principali caratteristiche di UserDefaults è la sua semplicità d'uso. Con poche righe di codice, è possibile salvare i dati nel UserDefaults utilizzando una chiave univoca per identificarli. Successivamente, è possibile recuperare i dati utilizzando la stessa chiave. Questo rende il processo di salvataggio e recupero dei dati molto intuitivo

Capitolo 3

Implementazione e descrizione delle funzionalità

In questo capitolo ci concentreremo sull'implementazione dell'applicazione GymZone descrivendo dettagliatamente le funzionalità e fornendo immagini per una migliore comprensione del progetto. Come già detto anche nei capitoli precedenti, per l'implementazione di GymZone sono state utilizzate le seguenti tecnologie:

1. Linguaggio di programmazione: GymZone è stata sviluppata utilizzando Swift come linguaggio di programmazione.
2. Framework di sviluppo: Per semplificare lo sviluppo e aggiungere nuove funzionalità, sono stati utilizzati i seguenti framework e librerie : SwiftUI, UserNotifications, Charts, CoreML, HealthKit.
3. Permanenza dei dati : GymZone garantisce la permanenza dei dati attraverso l'uso di UserDefaults che salva i dati su file.

L'applicazione GymZone si suddivide in diverse pagine, ognuna delle quali ha un ruolo specifico e svolge funzionalità diverse. Di seguito analizzeremo ognuna di queste, mostrando immagini e descrivendo i dettagli implementativi.

3.0.1 WorkoutsView

La sezione "Workouts" dell'applicazione è dedicata alla gestione e visualizzazione degli allenamenti dell'utente. La pagina presenta una lista di tutti gli allenamenti. Ogni allenamento è rappresentato come un elemento selezionabile all'interno di un 'DisclosureGroup', che mostra gli esercizi del workout con il rispettivo muscolo allenante quando viene espanso. E' possibile eliminare un allenamento dalla lista tramite uno swipe e conferma di eliminazione.(Figura 3.1)

Nella parte superiore della schermata è presente una barra di navigazione con titolo "My Workouts". Sono presenti anche due pulsanti nella barra di navigazione: il pulsante "dumbbell.fill" sulla sinistra per creare un allenamento standard e il pulsante "plus" sulla destra per aggiungere un nuovo allenamento personalizzato. L'allenamento standard creato è basato sul livello di allenamento dell'utente. sul sesso, sul peso e sull'altezza. Se si tenta di creare un nuovo allenamento standard senza aver completato il profilo utente con le informazioni necessarie (altezza, peso, sesso e livello di allenamento), verrà visualizzato un avviso (alert) che richiederà di completare il profilo

prima di procedere. Inoltre, se è già stata generata una scheda di allenamento standard, viene visualizzato un avviso che informa l'utente di eliminare la scheda precedente per crearne una nuova.(Figura 3.2 e 3.3).

Premendo il pulsante "plus" per aggiungere un nuovo allenamento, viene visualizzato un foglio (sheet) che ricopre la pagina principale con un'interfaccia per aggiungere i dettagli dell'allenamento, come il nome e gli esercizi da includere. Una volta completato, l'allenamento viene aggiunto alla lista degli allenamenti.

Inoltre, è presente una barra di navigazione inferiore con le altre pagine dell'applicazione: "Executions" per gestire le esecuzioni degli allenamenti, "Analytics" per effettuare una stima futura delle calorie bruciate e "Profile" per gestire le informazioni personali dell'utente.

In linea generale, la sezione "Workouts" dell'applicazione offre un'interfaccia intuitiva per la gestione e la visualizzazione degli allenamenti, consentendo all'utente di creare ed eseguire i propri programmi di allenamento in modo semplice e veloce.

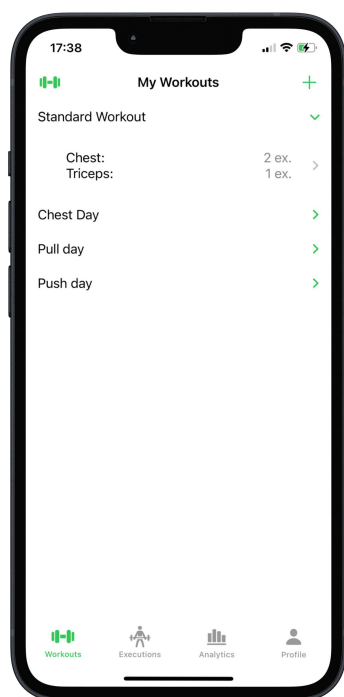


Figura 3.1: Workouts1

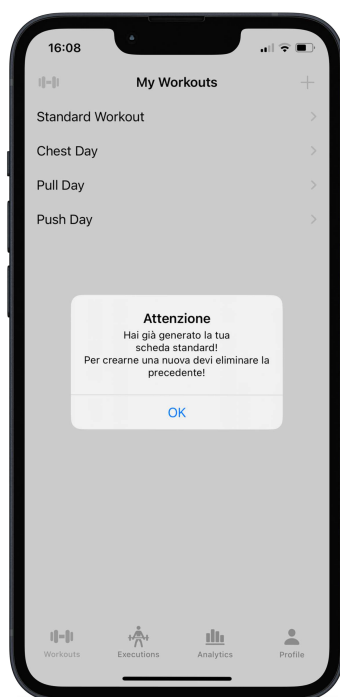


Figura 3.2: Workouts2



Figura 3.3: Workouts3

3.0.2 AddWorkoutView

La pagina "Add Workout" è dedicata alla creazione degli allenamenti. Consente agli utenti di personalizzare i propri allenamenti aggiungendo esercizi, impostando il nome dell'allenamento e programmando le notifiche per gli orari desiderati. (Figura 3.4)

All'interno di questa pagina, gli utenti possono inserire il nome personalizzato dell'allenamento desiderato in un campo di testo nella sezione "Name".

Nella sezione "Exercises", gli utenti possono visualizzare una lista degli esercizi aggiunti all'allenamento corrente. Questa lista viene dinamicamente aggiornata man mano che gli utenti aggiungono esercizi. Per ogni esercizio, vengono mostrate informazioni di base come il nome e la parte del corpo che allena.

Per impostare l'orario e il giorno dell'allenamento, gli utenti possono utilizzare la sezione "Training time". Qui possono selezionare il giorno della settimana desiderato e l'orario specifico in cui desiderano eseguire l'allenamento. Queste informazioni vengono utilizzate per programmare le notifiche degli allenamenti, in modo che gli utenti possano ricevere promemoria e rimanere motivati.

Infine, la pagina "Add Workout" offre i pulsanti di navigazione per l'aggiunta di esercizi e il salvataggio dell'allenamento. Cliccando su "Add Exercise", gli utenti possono accedere a una vista separata per aggiungere nuovi esercizi all'allenamento corrente. Cliccando su "Save", gli utenti possono salvare l'allenamento con tutte le informazioni e le impostazioni associate. Cliccando su Cancel è possibile annullare l'operazione e tornare alla pagina 'Workouts'.

In generale, la pagina "Add Workout" fornisce agli utenti un'interfaccia semplice da utilizzare per creare e personalizzare i propri allenamenti.

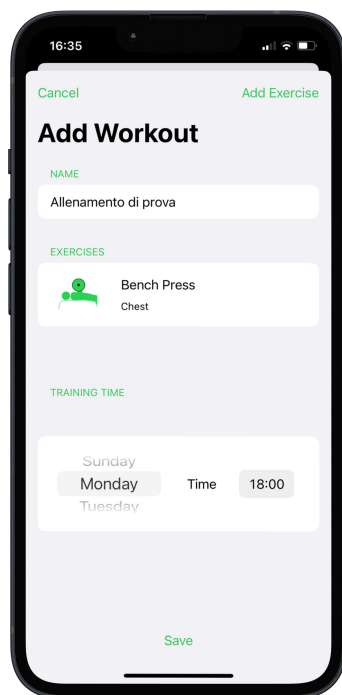


Figura 3.4: AddWorkout-View

3.0.3 AddExerciseView

La pagina "Add Exercise" permette agli utenti di cercare e selezionare esercizi da aggiungere al loro programma di allenamento. E' raggiungibile mediante il pulsante Add Exercise nella pagina 'Add Workout'.

E' formata da una barra di navigazione superiore che mostra il titolo "Exercises", e a sinistra un pulsante "Cancel" che permette agli utenti di annullare l'aggiunta dell'esercizio e tornare alla pagina precedente.

Successivamente è composta da una NavigationView contenente una lista di esercizi. Per ogni esercizio presente viene visualizzata una cella con il nome dell'esercizio, il muscolo allenante e una gif dimostrativa. Cliccando su una cella, gli utenti vengono reindirizzati alla pagina 'ExercisePropertyView' per visualizzare ulteriori dettagli sull'esercizio selezionato.

Inoltre, la pagina è dotata di una funzionalità di ricerca implementata tramite la modifica della variabile 'searchText'. Gli utenti possono digitare il nome dell'esercizio desiderato nella barra di ricerca per filtrare dinamicamente la lista degli esercizi. Infatti se 'searchText' è vuota (l'utente non ha inserito filtri) verrà restituito l'array completo contenente tutti gli esercizi, altrimenti, viene applicato un filtro all'array contenente gli esercizi. Ciò è implementato mediante il metodo filter che itera su ciascun elemento dell'array e verifica se il nome dell'esercizio contiene la stringa di ricerca. Il confronto viene effettuato convertendo sia il nome dell'esercizio che la stringa di ricerca in minuscolo, in modo da rendere la ricerca case-insensitive (senza distinzione tra maiuscole e minuscole). (Figura 3.6)

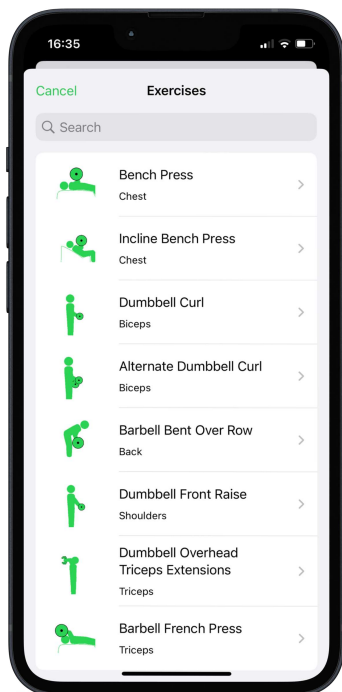


Figura 3.5: AddExercise-View 1

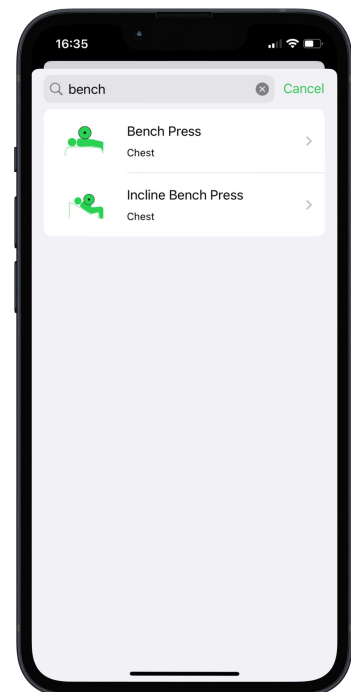


Figura 3.6: AddExercise-View 2

3.0.4 ExercisePropertyView

La vista 'ExercisePropertyView' permette di inserire le informazioni relative all'esercizio che si vuole aggiungere al piano di allenamento, come il numero di serie, il numero di ripetizioni e il tempo di recupero.

All'interno della vista, vengono utilizzate diverse variabili di stato(sets, reps, selectedMin, selectedSec) per tenere traccia dei valori inseriti dall'utente, tra cui il numero di serie, ripetizioni e il tempo di recupero. Queste variabili vengono collegate ai campi di testo e ai picker presenti nella vista tramite il binding '@State'.

La pagina contiene una NavigationView contenente un form . Dentro il Form, viene visualizzata una gif dimostrativa dell'esercizio. Successivamente è presente una sezione(sets) dedicata all'inserimento del numero di serie e ripetizioni dell'esercizio. Questa sezione mostra due campi di testo in cui l'utente può inserire i valori desiderati. Per assicurare la correttezza dell'input, i campi di testo sono configurati per utilizzare la tastiera numerica . Inoltre, sono applicati stili di design per rendere i campi di testo più gradevoli alla vista, come ad esempio l'arrotondamento degli angoli del textField.

La sezione successiva del form è dedicata all'inserimento del tempo di recupero (rest time). Qui sono presenti due picker, uno per i minuti e uno per i secondi. Tramite questi, l'utente può selezionare il valore desiderato. In aggiunta ai picker, vengono mostrati anche dei testi per indicare l'unità di misura del tempo selezionato.

Infine, la vista mostra il nome dell'esercizio corrente come titolo della barra di navigazione. Inoltre, è presente un pulsante di aggiunta ('Add') nella barra di navigazione che, quando premuto, richiama la funzione 'addExercise()'. Questa funzione si occupa di raccogliere i valori inseriti dall'utente dalle variabili di stato sopracitate, e di inserirli nell'esercizio del workout corrente che si sta creando.

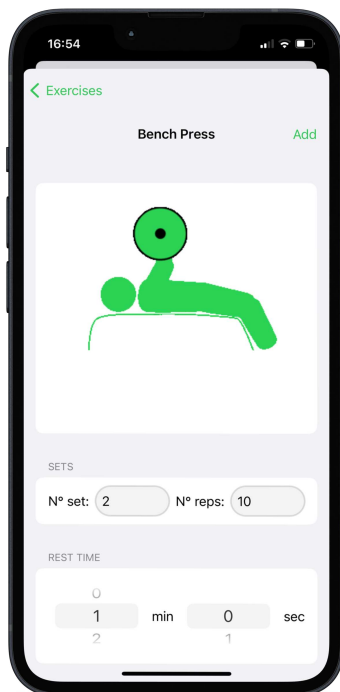


Figura 3.7: ExercisePropertyView

3.0.5 WorkoutView

La vista WorkoutView permette di visualizzare i dettagli di un allenamento specifico. E' accessibile dalla WorkoutsView, cliccando sul singolo allenamento.

All'interno di WorkoutView, viene mostrata la lista di esercizi che compongono l'allenamento. Ogni esercizio è rappresentato come un DisclosureGroup, che consente di espandere o comprimere i dettagli dell'esercizio. All'interno di ogni DisclosureGroup, vengono visualizzati i dettagli di ogni serie allenante, come il numero di ripetizioni, il peso e il tempo di recupero utilizzati nell'ultimo allenamento. Inoltre nella DisclosureGroup è presente un'istanza di ExerciseWorkoutCell, che presenta un'immagine rappresentativa dell'esercizio, il nome dell'esercizio stesso e altre informazioni utili come la parte del corpo coinvolta e il numero di set.

In alto è presente la barra di navigazione in cui è possibile avviare l'allenamento o tornare alla WorkoutsView. Infatti attraverso delle variabili di stato è controllata l'interazione dell'utente con l'applicazione. Se l'utente clicca su play, la variabile 'isPlay' viene settata e permette l'inizio dell'allenamento passando alla PlayView mediante un foglio a schermo intero che si sovrappone all'attuale vista.

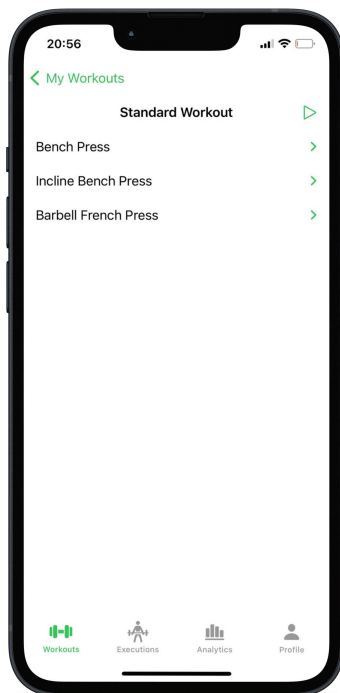


Figura 3.8: WorkoutView1

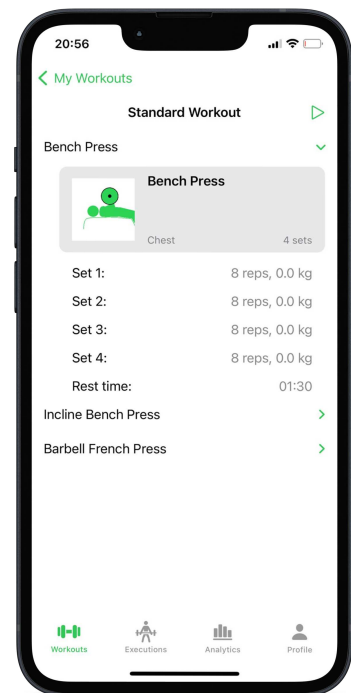


Figura 3.9: WorkoutView2

3.0.6 PlayView

La PlayView rappresenta l'interfaccia durante l'esecuzione di un workout, permettendo agli utenti la possibilità di inserire i risultati delle serie allenanti e passare agli esercizi successivi.

All'avvio della vista, viene mostrata una gif rappresentativa dell'esercizio in corso. Sotto di essa, è presente un timer visuale che indica il tempo di riposo tra le serie dell'esercizio. Il timer può essere avviato o resettato mediante gli appositi pulsanti.

L'utente può inserire i risultati delle serie nella sezione "Sets". Per ciascuna serie, è presente un campo di testo in cui l'utente può inserire il numero di ripetizioni e il peso utilizzato. I dati inseriti vengono salvati nella variabile di stato 'newWeight'. Se l'utente ha già eseguito quell'allenamento, vedrà nel campo di testo dove si inserisce il peso, il valore del peso utilizzato nell'allenamento precedente, in modo da poter provare a migliorare la sua performance aumentando il peso (figura 3.12).

Se è la prima volta che esegue quel workout vedrà il valore del textField settato a '0' (figura 3.11).

L'utente può passare all'esercizio successivo premendo il pulsante "Next" nella barra inferiore della vista. Infatti alla pressione del bottone si attiva la funzione nextset() in cui i dati dell'esercizio corrente vengono salvati nell'oggetto d'ambiente workoutStore, che rappresenta l'insieme dei workout, condiviso per tutte le viste dell'applicazione. Successivamente, se l'esercizio corrente non è l'ultimo del workout la vista si aggiorna per visualizzare i dettagli del nuovo esercizio. Se l'esercizio corrente è l'ultimo del workout viene presentata una vista a schermo intero 'AwardView', che mostra un riepilogo dell'allenamento appena completato. Questa vista permette all'utente di visualizzare i risultati del workout e di condividere eventuali obiettivi raggiunti.

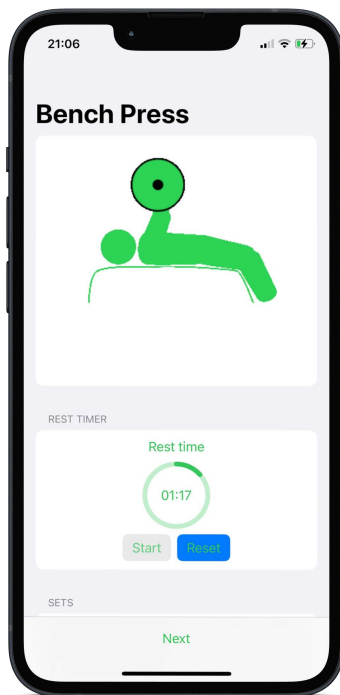


Figura 3.10: PlayView1

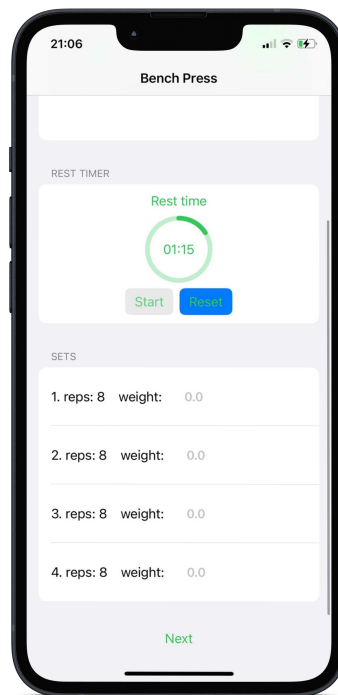


Figura 3.11: PlayView2

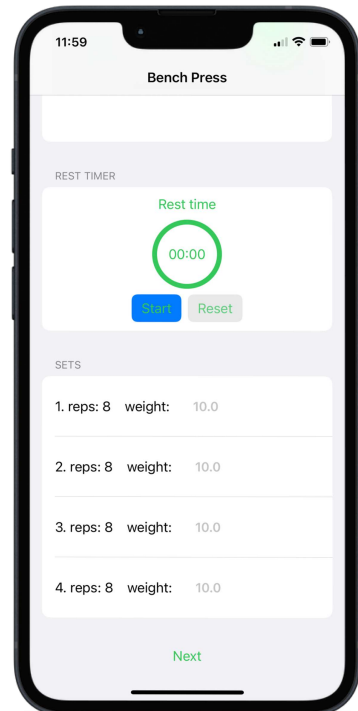


Figura 3.12: PlayView3

Analisi della funzione 'nextset'

```
1 func nextset() {  
2     for index in 0..3         exerciseindex].sets.count {  
4         if newWeight[index] == "" {  
5             newWeight[index] = "0"  
6         }  
7         workoutStore.workouts[workoutindex].exercises[exerciseindex].  
8             sets[index].weight = Double(newWeight[index])!  
9     }  
10  
11     if workoutStore.workouts[workoutindex].exercises.count-1 ==  
12         exerciseindex {  
13         isFinalView.toggle()  
14     } else {  
15         isNextView.toggle()  
16     }  
17     DataManager.shared.saveWorkouts(workoutStore.workouts)  
18 }
```

Listing 3.1: Funzione nextset

Nextset è la funzione che viene richiamata quando l'utente preme il pulsante 'Next' all'interno della PlayView. Ha lo scopo di far scorrere tutti gli esercizi presenti all'interno dell'allenamento che l'utente sta eseguendo.

Riportiamo una descrizione più dettagliata della funzione:

1. Per prima cosa, viene effettuato un ciclo for che si ripete tante volte quante sono il numero di serie associate all'esercizio corrente.
2. All'interno del ciclo, viene controllato se il valore di newWeight[index], che rappresenta il peso inserito associato a quell'esercizio, è vuoto (""). In questo caso, il valore di newWeight[index] viene impostato a 0. Questo controllo è effettuato per gestire i casi in cui l'utente non inserisce un peso per una determinata serie.
3. Successivamente, viene aggiornato il valore del peso utilizzato per quella serie all'interno del workout che si sta eseguendo.
4. Dopo aver completato il ciclo for, viene effettuato un controllo per verificare se l'esercizio corrente è l'ultimo nell'array delle serie. Questo viene fatto confrontando l'indice exerciseindex, che inizialmente è 0 e viene passato di volta in volta quando si richiama la PlayView incrementandolo di uno, con il numero di esercizi associato a quel piano di allenamento. In tal caso, viene attivata la variabile di stato isFinalView, che permette di visualizzare la vista WorkoutViewAward a schermo intero.
5. Se l'esercizio corrente non è l'ultimo, viene attivata la variabile di stato isNextView, che permette di passare alla vista successiva dell'esercizio durante l'esecuzione del workout.
6. Infine, viene chiamata la funzione saveWorkouts dell'oggetto DataManager.shared per salvare i workout aggiornati nell'archivio dei dati e garantire la permanenza dei dati.

3.0.7 AwardView

La pagina 'AwardView' rappresenta l'interfaccia utente per visualizzare il termine di un allenamento e viene visualizzata quando l'utente termina l'esecuzione di un workout. L'obiettivo principale di questa pagina è fornire agli utenti una panoramica dettagliata degli esercizi svolti durante l'allenamento.

La pagina è strutturata come una vista di navigazione, che permette agli utenti di visualizzare gli esercizi e tornare alla schermata precedente mediante il pulsante indietro. All'interno della vista di navigazione è presente una lista contenente tutti gli esercizi dell'allenamento. Per ogni esercizio, è utilizzata la vista 'ExerciseWorkoutCellAward' che mostra l'immagine dell'esercizio, il nome, la parte del corpo coinvolta, il numero di set, il numero di ripetizioni e il peso utilizzato per ogni ripetizione.

La pagina 'AwardView' utilizza anche alcuni effetti speciali per migliorare l'user experience dando all'utente un feedback visivo relativo al termine dell'allenamento. In particolare sono implementati due cerchi di colore tramite un modifier chiamato 'ParticlesModifier' che forniscono un effetto animato di fuochi d'artificio intorno alla pagina (figura 3.13).

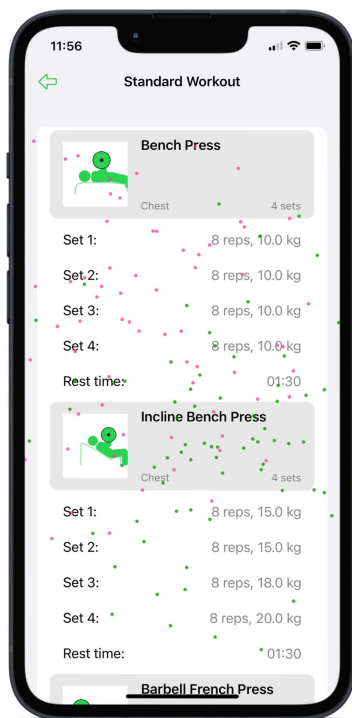


Figura 3.13: AwardView1

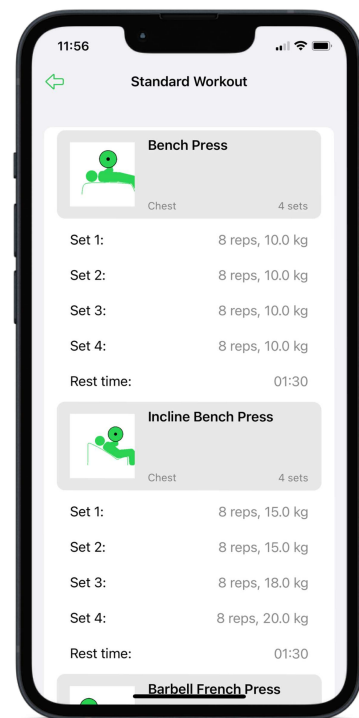


Figura 3.14: AwardView2

3.0.8 ExecutionView

La sezione 'ExecutionView' dell'applicazione ha il compito far visualizzare e di descrivere le esecuzioni degli esercizi.

All'interno della vista, è presente un campo di ricerca in cui è possibile inserire del testo per filtrare gli esercizi che si vogliono visualizzare in base al testo inserito dall'utente. Ovviamente se il campo di ricerca è vuoto, vengono mostrati tutti gli esercizi disponibili all'interno dell'applicazione.

La vista contiene una 'NavigationView' per permettere la navigazione tra le diverse schermate. All'interno di essa, è presenta una lista per visualizzare gli esercizi filtrati. Ciascun esercizio viene rappresentato tramite la struttura 'ExerciseStaticCell'. Questa rappresenta una cella che mostra la gif dell'esercizio, il nome e la parte del corpo a cui appartiene.

Cliccando su un esercizio, l'utente viene reindirizzato alla vista 'SingleExecutionView'. Tale vista mostra i dettagli completi dell'esercizio, fornendo una descrizione e una gif che mostra l'esecuzione.

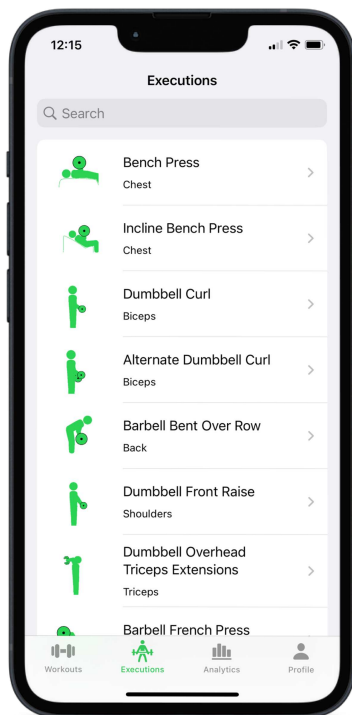


Figura 3.15: Execution-View1

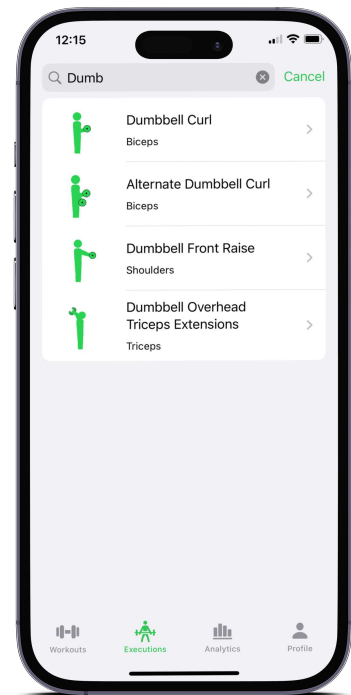


Figura 3.16: Execution-View2

3.0.9 SingleExecutionView

La vista 'SingleExecutionView' dell'applicazione permette di far visualizzare all'utente una gif dimostrativa dell'esercizio e una descrizione accurata sull'esecuzione di quest'ultimo. In questo modo l'utente dell'applicazione può accertarsi dell'esecuzione corretta di un esercizio che fa parte del suo allenamento.

All'interno della vista, viene utilizzato un `NavigationView` che permette di spostarsi tra le schermate dell'applicazione in modo intuitivo.

Il contenuto principale della vista è racchiuso in un `Form`, che organizza gli elementi all'interno di sezioni. All'interno del `Form`, viene visualizzata una gif che mostra la corretta esecuzione dell'esercizio .

Inoltre, è presente una sezione dedicata alla descrizione dell'esercizio, evidenziata con un titolo "Exercise description".

Infine, il titolo della navigazione viene impostato come il nome dell'esercizio. Questo consente all'utente di avere un riferimento chiaro all'esercizio che sta visualizzando mentre naviga tra le schermate dell'applicazione.

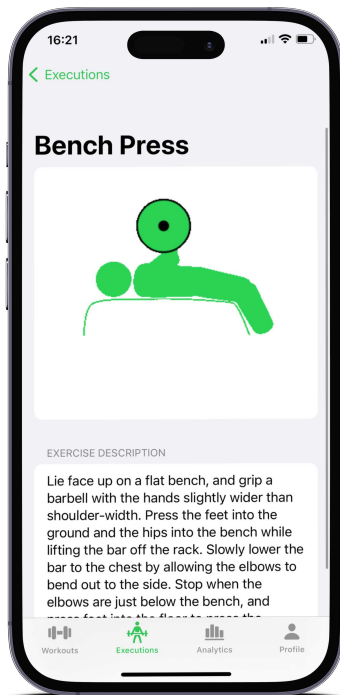


Figura 3.17: SingleExecutionView

3.0.10 ProfileView

Nell'interfaccia 'ProfileView' gli utenti possono visualizzare e gestire il proprio profilo. Salvando i propri dati in questa pagina, gli utenti possono anche confrontarli con dati futuri, per analizzare eventuali miglioramenti dovuti al loro percorso di allenamento.

La vista contiene una 'NavigationView' per permettere la navigazione tra le diverse schermate. All'interno di essa, sono presenti diverse sezioni organizzate all'interno di un Form. La prima sezione riguarda le "Informazioni personali" dell'utente, come il nome, il sesso e il livello di esperienza nel fitness. Gli utenti possono inserire o modificare queste informazioni utilizzando campi di testo interattivi o menu a discesa.

La sezione successiva si concentra sulle "Misure antropometriche", consentendo agli utenti di inserire i loro dati relativi all'altezza e al peso. Inoltre, è presente un bottone nel campo di testo, per visualizzare un grafico che mostra la variazione del peso nel tempo.

Se l'utente seleziona l'opzione per visualizzare il grafico del profilo, viene aperta una schermata a schermo intero che mostra il grafico completo utilizzando la vista 'Chart-View'.

Infine, la sezione delle "Circonferenze" permette agli utenti di inserire le misure di diverse parti del corpo, come collo, spalle, bicipiti, tricipiti, avambracci e polpacci. Questi dati vengono inseriti tramite campi di testo e permettono all'utente di tener traccia di eventuali miglioramenti nel tempo.

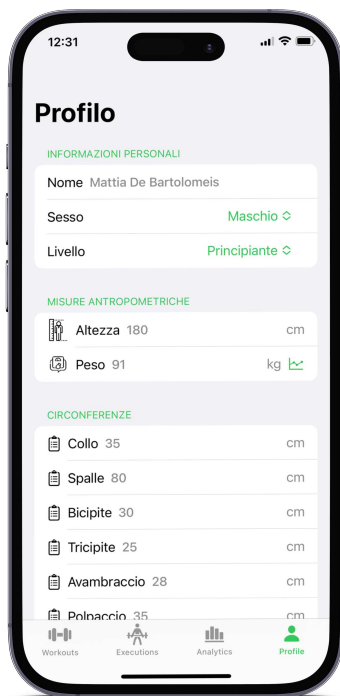


Figura 3.18: ProfileView1

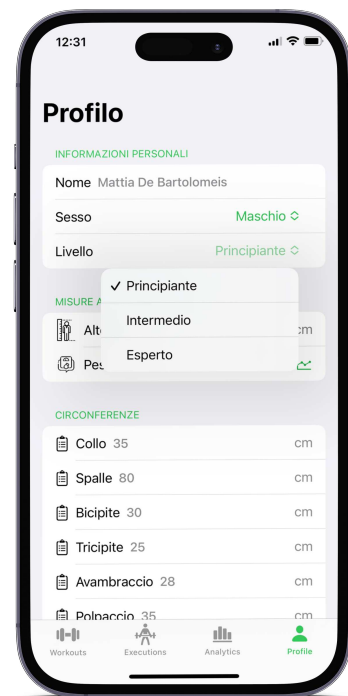


Figura 3.19: ProfileView2

3.0.11 ChartView

La pagina 'ChartView' consente agli utenti di visualizzare l'andamento del loro peso nel corso del tempo. Sfruttando la libreria Charts di Swift, la vista offre una rappresentazione grafica del peso dell'utente.

All'interno di questa pagina, i dati del peso vengono prelevati dall'oggetto 'Profilo', che, essendo dichiarato come variabile ambientale, è condiviso nell'ambiente di esecuzione dell'applicazione. Questi dati vengono quindi visualizzati tramite un grafico lineare che presenta sull'asse x la data relativa al peso e sull'asse y i valori del peso. Il colore della linea del grafico è personalizzato in verde per una migliore leggibilità. Il grafico è personalizzato utilizzando le funzionalità offerte dalla libreria Charts. Possiamo infatti notare un bordo sottile e uno sfondo bianco per evidenziare al meglio i dati. Inoltre, per bilanciare al meglio il grafico, la scala sull'asse Y del grafico è impostata in modo dinamico adattandosi al valore massimo e minimo del peso dell'utente.

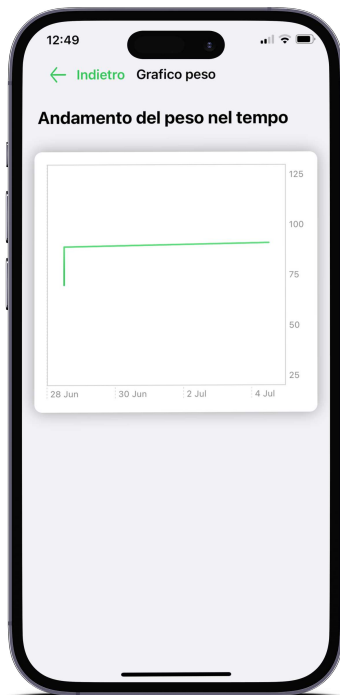


Figura 3.20: ChartView

3.0.12 AnalyticsView

La pagina "AnalyticsView" permette all'utente di inserire una stima dei dati relativi all'attività fisica che svolgerà durante i prossimi tre giorni . I dati includono il numero di passi, i minuti trascorsi inattivi, i minuti trascorsi in attività fisica e la distanza totale percorsa.

La vista è organizzata attraverso un "Form" , che suddivide i dati dell'utente in sezioni per ciascun giorno. Ogni sezione contiene campi di testo in cui l'utente può inserire i propri dati.

Alla pressione del pulsante "Save", i dati inseriti dall'utente vengono passati alla funzione "passToButton()". Questa funzione effettua una stima delle calorie che brucerà l'utente utilizzando un modello di machine learning implementato con il framework CoreML. I risultati di tale stima vengono poi visualizzati nella vista 'Prediction View'. La vista "AnalyticsView" si occupa anche di richiedere l'autorizzazione all'accesso ai dati presenti nell'app 'Salute' di apple , utilizzando il framework HealthKit. Infatti tali dati saranno utilizzati sempre nella vista 'Prediction View' in cui sarà visualizzato un grafico che metterà a confronto le calorie bruciate nei giorni passati, ricavate dall'app 'Salute' con la stima delle calorie che l'utente brucerà nei prossimi tre giorni.

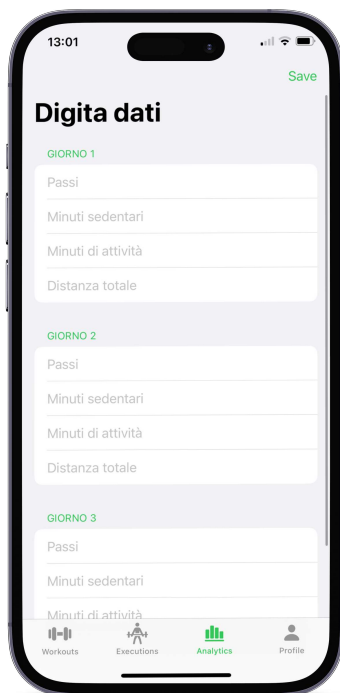


Figura 3.21: AnalyticsView

NOTA: Desidero precisare che questa View non è stata realizzata da me, ma da un collega che ha collaborato nello sviluppo dell'applicazione.

3.0.13 PredictionView

La pagina 'PredictionView' si occupa di mostrare all'utente le previsioni delle calorie bruciate nei prossimi tre giorni confrontandole con le calorie bruciate nei giorni

precedenti.

Il cuore della vista è il grafico delle previsioni delle calorie, generato utilizzando la libreria 'Charts'. Il grafico rappresenta le calorie reali bruciate e le calorie previste, consentendo all'utente di confrontare visivamente i dati. Le calorie reali vengono mostrate come barre verticali blu, mentre le calorie previste vengono visualizzate come barre verticali verdi.

La vista offre anche delle funzionalità aggiuntive. Ad esempio, è possibile condividere l'immagine del grafico con altre persone o salvarla nella libreria delle foto. Questo permette all'utente di tenere traccia delle proprie previsioni di calorie bruciate nel tempo e di condividerle con i propri amici o professionisti del settore del fitness.

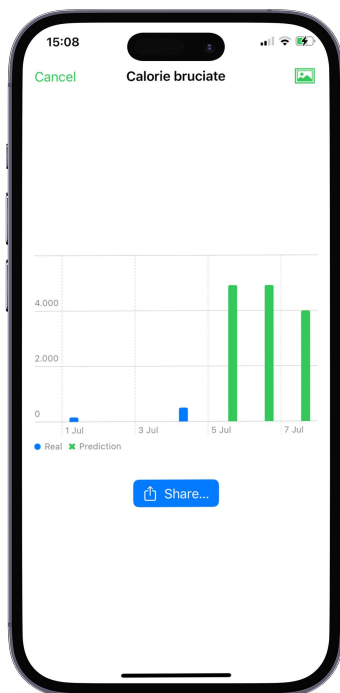


Figura 3.22: PredictionView

NOTA: Desidero precisare che questa View non è stata realizzata da me, ma da un collega che ha collaborato nello sviluppo dell'applicazione.

3.0.14 Gestione delle notifiche

Come accennato anche nei capitoli precedenti, è stato implementato un sistema di gestione delle notifiche locali mediante l'utilizzo della libreria UserNotifications. In particolare quando l'utente aggiunge un workout al proprio workoutStore, inserirà un'orario e un giorno di allenamento. Utilizzando questi dati, l'utente riceverà una notifica di promemoria per ricordarsi di svolgere la sessione di allenamento programmata. La notifica verrà inviata all'orario prestabilito per avvisare l'utente che è arrivato il momento di iniziare la sua sessione di allenamento. L'implementazione di tale sistema di notifiche è stato implementato all'interno di una classe apposita, che presenta due metodi principali:

1. Il metodo "requestAuthorization" viene utilizzato per richiedere l'autorizzazione dell'utente a ricevere notifiche.
2. Se l'utente rilascia l'autorizzazione per ricevere notifiche dall'applicazione, viene utilizzato il metodo "scheduleNotification" per schedulare una notifica. Quest'ultimo riceve una data di riferimento, il nome dell'utente e il nome del workout associato alla notifica. Durante la schedulazione della notifica, viene impostato il nome del workout come 'identificatore', in modo tale che quando l'utente rimuoverà il workout dal suo store, potremo eliminare anche la notifica associata sfruttando l'identificatore.

Infine quando un utente elimina un allenamento, verrà automaticamente eliminata anche la relativa notifica associata. Ciò è reso possibile dall'attribuzione del nome dell'allenamento come identificativo per la notifica programmata. Pertanto, durante la funzione di eliminazione dell'allenamento, viene anche invocata la funzione di userNotifications, che utilizzando l'identificativo della notifica, procede alla sua rimozione dalla pianificazione.

Di seguito sono riportati due esempi di notifica. Se l'utente non ha inserito il proprio nome nella sezione profilo, la notifica non includerà nel testo il nome dell'utente (figura 3.23).

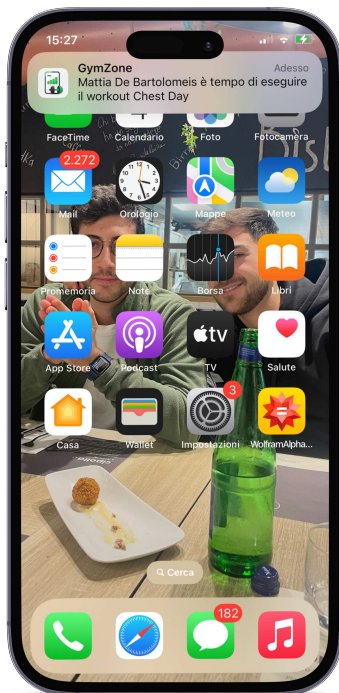


Figura 3.23: Notifica 1

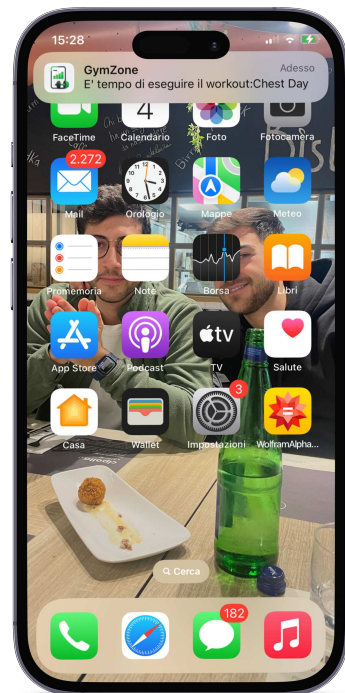


Figura 3.24: Notifica 2

Capitolo 4

Conclusioni e prospettive future

In questo capitolo, presenteremo una sintesi del lavoro sviluppato per il progetto GymZone, fornendo spunti di miglioramento e possibili evoluzioni in relazione agli argomenti trattati.

Durante il percorso di sviluppo dell'applicazione GymZone, la soluzione a cui si puntava, come descritto in fase di progettazione, era la seguente: "Un'applicazione su cui organizzare e monitorare l'allenamento delle persone permettendo di tracciare il peso e le ripetizioni, evidenziando i progressi ottenuti e fornendo istruzioni precise per l'esecuzione corretta di ogni esercizio".

Durante l'implementazione di GymZone, sono stati raggiunti importanti traguardi, essendo state sviluppate tutte le funzionalità principali dell'applicazione. Gli utenti infatti, utilizzando GymZone, possono:

1. Organizzare i propri allenamenti: GymZone permette attraverso un interfaccia intuitiva di creare il proprio allenamento, inserendo gli esercizi che si desidera eseguire e salvarlo direttamente all'interno dell'applicazione.
2. Tener traccia dei loro progressi: GymZone offre ai suoi utenti diversi modi per tenere traccia dei loro progressi fisici e delle performance. Nella sezione del profilo, è possibile monitorare dati importanti come il peso corporeo e le misure muscolari. Inoltre, viene fornito un grafico che illustra l'andamento del peso nel corso del tempo. Ogni sessione di allenamento registra anche i pesi utilizzati nelle sessioni precedenti, consentendo agli utenti di notare facilmente eventuali miglioramenti durante le loro sessioni di allenamento.
3. Eseguire correttamente gli esercizi: GymZone offre ai suoi utenti la possibilità di eseguire gli esercizi dell'allenamento in modo corretto attraverso una sezione dedicata chiamata "Esecuzioni". In questa sezione, gli utenti possono trovare una vasta gamma di esercizi disponibili nell'applicazione, accompagnati da esempi dimostrativi in forma di gif e descrizioni dettagliate dell'esecuzione. Questo permette agli utenti di apprendere correttamente la tecnica di ogni esercizio, evitando potenziali errori e riducendo il rischio di infortuni durante l'allenamento.

Inoltre sono state implementate ulteriori funzionalità 'minori' che contribuiscono a rendere l'applicazione più completa. E' possibile infatti ottenere una scheda standardizzata, ottenere notifiche di promemoria per l'allenamento e ottenere una stima delle calorie bruciate in futuro. Nonostante ciò, ci sono ancora ampi spazi per il miglioramento e

per future evoluzioni dell'applicazione. Alcuni suggerimenti per il futuro sviluppo di GymZone includono:

1. Miglioramento dell'interfaccia utente: Si potrebbe lavorare per migliorare l'esperienza dell'utente attraverso un'interfaccia più intuitiva e una grafica più accattivante. L'obiettivo è quello di rendere l'applicazione ancora più facile da usare e visivamente più gradevole.
2. Calendarizzazione degli allenamenti: Si potrebbe implementare una funzione che sincronizzi gli allenamenti dell'utente direttamente al calendario del dispositivo dell'utente, sfruttando il framework EventKit. In questo modo l'utente può avere una panoramica chiara di tutti i suoi impegni giornalieri, settimanali e mensili e ricevere notifiche di promemoria personalizzabili direttamente da Calendar.
3. Integrazione con Apple Watch : Si potrebbe valutare la possibilità di integrare GymZone con l'Apple Watch, al fine di migliorare nettamente l'esperienza dell'utente durante l'esecuzione dell'allenamento. Infatti Sarebbe decisamente più pratico utilizzare direttamente l'orologio al polso invece del telefono per registrare i pesi utilizzati durante l'allenamento e gestire i tempi di recupero. Un esempio di interfaccia grafica di GymZone sull'apple watch potrebbe essere la seguente.



Figura 4.1: Watch

4. Espansione della base di dati tramite l'utilizzo di un sistema di gestione di database (DBMS): Si potrebbe considerare l'aggiunta di una più ampia gamma di esercizi e programmi di allenamento all'interno dell'applicazione, al fine di offrire agli utenti un'esperienza ancora più completa. In particolare sarebbe ideale l'implementazione di un database per la gestione dei dati, in quanto migliorerebbe l'efficienza complessiva del sistema e fornirebbe uno spazio disponibile maggiore per memorizzare le informazioni.
5. Interfacciamento con il coach: Si potrebbe implementare una funzione che consenta all'utente di interfacciarsi con il proprio allenatore, o con allenatori che utilizzano l'applicazione. L'interfacciamento con il coach potrebbe avvenire tramite una chat integrata nell'applicazione o attraverso videochiamate, consentendo una comunicazione diretta tra l'utente e il coach. Attraverso questa funzionalità, gli utenti potrebbero richiedere consigli su esercizi specifici, chiedere suggerimenti per la pianificazione dell'allenamento o addirittura condividere i propri progressi con il coach.

Guardando al futuro, l'applicazione GymZone ha il potenziale per diventare un punto di riferimento nel campo dell'allenamento personale. Continuare a lavorare sulla sua evoluzione e implementare le funzionalità suggerite consentirà agli utenti di raggiungere nuovi livelli di personalizzazione e coinvolgimento nel loro percorso di fitness.

È importante sottolineare che GymZone non è solo uno strumento tecnologico, ma anche un supporto motivazionale per gli utenti. La possibilità di monitorare i progressi, ricevere consigli dagli allenatori e condividere le proprie esperienze con una comunità attiva potrà avere un impatto positivo sulla loro motivazione e successo nel raggiungere gli obiettivi personali di fitness.

Bibliografia e sitografia

- [1] Apple. *MacBook Pro*. URL: https://support.apple.com/kb/SP824?locale=it_IT.
- [2] Apple. *iPhone 13*. URL: <https://www.apple.com/it/iphone-13/specs/>.
- [3] Apple. *XCode*. URL: <https://developer.apple.com/xcode/>.
- [4] Apple. *Swift*. URL: <https://developer.apple.com/swift/>.
- [5] Apple. *SwiftUI*. URL: <https://developer.apple.com/xcode/swiftui/>.
- [6] Apple. *UserNotifications Swift*. URL: <https://developer.apple.com/documentation/usernotifications>.
- [7] Apple. *Charts Swift*. URL: <https://developer.apple.com/documentation/charts>.
- [8] Apple. *CoreML*. URL: <https://developer.apple.com/documentation/coreml>.
- [9] Apple. *HealthKit*. URL: <https://developer.apple.com/documentation/healthkit>.
- [10] Apple. *Challenge Based Learning*. URL: https://www.apple.com/br/education/docs/CBL_Classroom_Guide_Jan_2011.pdf.
- [11] Apple. *Human Interface Guidelines*. URL: <https://developers.apple.com/design/human-interface-guidelines/guidelines/overview/>.
- [12] Marco Romano. *UX concepts*.

Elenco delle figure

| | | |
|------|--------------------------------|----|
| 2.1 | Schermo chiaro | 17 |
| 2.2 | Schermo scuro | 17 |
| 3.1 | Workouts1 | 21 |
| 3.2 | Workouts2 | 21 |
| 3.3 | Workouts3 | 21 |
| 3.4 | AddWorkoutView | 22 |
| 3.5 | AddExerciseView 1 | 23 |
| 3.6 | AddExerciseView 2 | 23 |
| 3.7 | ExercisePropertyView | 24 |
| 3.8 | WorkoutView1 | 25 |
| 3.9 | WorkoutView2 | 25 |
| 3.10 | PlayView1 | 26 |
| 3.11 | PlayView2 | 26 |
| 3.12 | PlayView3 | 26 |
| 3.13 | AwardView1 | 28 |
| 3.14 | AwardView2 | 28 |
| 3.15 | ExecutionView1 | 29 |
| 3.16 | ExecutionView2 | 29 |
| 3.17 | SingleExecutionView | 30 |
| 3.18 | ProfileView1 | 31 |
| 3.19 | ProfileView2 | 31 |
| 3.20 | ChartView | 32 |
| 3.21 | AnalyticsView | 33 |
| 3.22 | PredictionView | 34 |
| 3.23 | Notifica 1 | 35 |
| 3.24 | Notifica 2 | 35 |
| 4.1 | Watch | 37 |