Benessere

Versione: 1.0.0a

Data di rilascio: 01/02/2021

**Realizzato da**

Fabio Resta

Enrico Masellis

Davide Palladino

Magrone Giovanni

Mauro Marzocca

Indice

1. Product Backlog 3

1.1 Introduzione 3

1.2 Contesto di business 3

1.3 Stakeholder 3

1.3.1 Scrum Master 3

1.3.2 Product Owner 3

1.3.3 Developer 3

1.4 Item funzionali 3

1.5 Item informativi 6

1.6 Tecnologie utilizzate 6

3. Glossario 8

3.1 Definizioni 8

3.2 Istruzioni operative 8

# Product Backlog

## Introduzione

Benessere è un’applicazione che permette la gestione di diete e schede di allenamento inserite dai relativi professionisti, come dietologi e coach.

Un utente, dopo aver contattato il proprio professionista e avergli comunicato il suo username, può ricevere la sua dieta o scheda di allenamento tramite l’applicazione. Gli viene mostrato il rispettivo documento diviso per giorni, dal lunedì alla domenica.

Un utente può scegliere di inviare una richiesta di modifica se un alimento della dieta non gli piace. La richiesta viene inviata al dietologo, che può decidere se approvarla. Appena la richiesta è approvata, la dieta viene immediatamente aggiornata.

La struttura della dieta è flessibile, il dietologo può inserire alimenti in base al giorno, tipo di pasto e porzioni. Tutte queste caratteristiche possono essere modificate in seguito, cliccando sull’alimento da modificare nella lista.

Se l’utente non ha comprato gli alimenti presenti nella sua dieta può usare la lista della spesa integrata nell’applicazione. Con essa può inserire, modificare e cancellare gli elementi della lista, oltre a spuntare quelli acquistati.

Un utente può anche seguire una scheda di allenamento, gestita dal suo coach.

Per allenarsi all’aria aperta l’utente può usare il contapassi integrato nell’applicazione per svolgere passeggiate e corsette.

L’utente può accedere facilmente alla palestra aprendone i tornelli wireless utilizzando l’applicazione.

## Contesto di business

Benessere vuole essere una soluzione per dietologi e coach.

Benessere è diversa dalle altre applicazioni di dieta e fitness perché permette agli esperti di interfacciarsi con i propri clienti e gestire in remoto le operazioni di routine, come la modifica della dieta.

Il cliente può invece accedere alla dieta e alla scheda di allenamento direttamente dall’app, eliminando la necessità di conservare fogli e altri documenti in casa.

## Stakeholder

### Scrum Master

Fabio Resta

### Product Owner

Fabio Resta

### Developer

Fabio Resta

Enrico Masellis

Davide Palladino

Magrone Giovanni

Mauro Marzocca

### Design UI e grafica

Giovanni Magrone

Mauro Marzocca

### Gestione fragment e sensore contapassi

Davide Palladino

### Gestione DB (CRUD) e modellazione

Fabio Resta

Enrico Masellis

### Registrazione e login

Giovanni Magrone

### Apertura tornello

Davide Palladino

## Item funzionali

### IF-1 Registrazione e login

Come utente voglio registrarmi e accedere all’applicazione così che possa utilizzare i servizi dell’applicazione.

Il login e la registrazione dell’app sono divisi tra esperti (dietologo/coach) e utenti normali.

### IF-2 Inserire un cliente

Come dietologo o coach voglio inserire un cliente.

Dietologi e coach possono inserire i clienti, consultando il database di clienti registrati.

Quando un utente svolge una visita da dietologo o si reca in palestra, può comunicare al professionista di voler usare l’applicazione. Il professionista può cercare il cliente tramite username univoco.

### IF-3 Inserire una dieta

Come dietologo voglio inserire una dieta per un cliente.

Il dietologo, dopo aver aggiunto un suo cliente cercandolo per username, può inserire la dieta specificando tutti i parametri necessari, come tipologia del pasto, giorno e porzione. Se commette un errore può premere sul pasto e modificarlo. Viene utilizzata la stessa activity sia per l’inserimento che per la modifica o eventuale cancellazione del pasto, mostrando solo i pulsanti necessari.

### IF-4 Visualizzare una dieta

Come cliente voglio visualizzare la mia dieta settimanale.

Il cliente, dopo aver contattato il suo dietologo, può ricevere una dieta settimanale. L’utente può premere su un alimento della dieta per inviare una richiesta di modifica.

Per realizzare una dieta si è deciso di creare un fragment con 7 tab, una per ogni giorno della settimana.

### IF-5 Inviare una richiesta di modifica alla dieta

Come utente voglio inviare una richiesta al mio dietologo.

Un utente può inviare una richiesta di modifica della dieta se non ha un particolare alimento ecc.

Per inviare una modifica è sufficiente cliccare su un alimento, verrà mostrato un dialog fragment con cui compilare la richiesta e inviarla. La richiesta dovrà poi essere approvata dal dietologo

### IF-6 Approvare o modificare una richiesta di modifica alla dieta

Come dietologo voglio approvare o modificare le richieste dei clienti.

Il dietologo può ricevere richieste inviate dai suoi clienti. Quando apre una richiesta può approvarla oppure inserire ulteriori modifiche.

### IF-7 Gestione della lista della spesa

Come cliente voglio gestire la mia lista della spesa.

Ogni cliente iscritto all’applicazione, nella sezione dieta, ha una lista della spesa personale su cui inserire i prodotti che deve comprare. La lista della spesa supporta operazioni CRUD.

Per realizzarla, è stata implementata una RecyclerView con adapter personalizzato. Tutte le informazioni sono poi salvate su SQLite

### IF-8 Inserire una scheda di allenamento

Come coach voglio inserire una scheda di allenamento per un cliente.

Il coach, dopo aver aggiunto un suo cliente cercandolo per username, può inserire la scheda di allenamento specificando tutti i parametri necessari, come tipologia dell’esercizio, giorno e ripetizioni. Se commette un errore può premere sull’esercizio e modificarlo. Viene utilizzata la stessa activity sia per l’inserimento che per la modifica o eventuale cancellazione dell’esercizio, mostrando solo i bottoni necessari.

### IF-9 Visualizzare la scheda di allenamento

Come cliente voglio gestire la mia scheda di allenamento.

Il cliente può visualizzare la propria scheda di allenamento personale.

Diversamente dalla dieta l’utente non può inviare richieste di modifica.

### IF-10 Ricevere notifiche

Come cliente voglio ricevere notifiche su pasti.

Quando l’utente accede all’applicazione, sono automaticamente schedulate delle notifiche che gli ricorderanno l’orario dei pasti principali, pranzo e cena.

Per realizzare questa funzionalità si è utilizzato un notificationHelper per gestire le notifiche e creare canali e un AlarmReceiver per schedulare le notifiche al giusto orario.

### IF-11 Accedere facilmente alla palestra

Come cliente voglio accedere facilmente alla palestra, aprendo il tornello di accesso in maniera wireless.

Per realizzare questa funzionalità abbiamo usato la connettività wifi. L’applicazione invia un messaggio tramite socket ad un microcontrollore ESP-32. Il microcontrollore riceverà il messaggio e azionerà il motorino del tornello. Per testare questa funzionalità è necessario avere un ESP-32 con installato il codice, inserito nella cartella Platform-IO del progetto.

### IF-12 Personalizzare il mio profilo

Come cliente voglio personalizzare il mio profilo, aggiungendo foto e generalità.

Un cliente che utilizza Benessere può inserire una sua foto e modificare alcuni parametri del suo profilo. Per la funzionalità fotocamera, sono stati gestiti i permessi necessari e le foto sono salvate nella cartella privata dell’applicazione.

### IF-13 Conoscere la funzionalità degli attrezzi

Come cliente voglio conoscere la funzionalità degli attrezzi.

Benessere contiene un piccolo glossario che spiega all’utente le funzionalità degli attrezzi usati in palestra. Questa funzionalità è stata sviluppata con i fragment, approccio master-detail

### IF-14 Allenarmi all’aria aperta

Come cliente voglio allenarmi all’aria aperta.

E’ stato implementato il sensore contapassi, con esso il cliente può decidere di fare una passeggiata o una corsetta all’aria aperta. Cliccando sull’icona del sole nell’actionbar l’utente riceverà gli aggiornamenti meteo, che gli consiglieranno se uscire o meno.

## Item informativi

### II-1 Tempestività

Il sistema deve essere completato entro il primo appello di febbraio.

### II-2 Correttezza

Il sistema deve soddisfare tutti i requisiti funzionali

## Tecnologie utilizzate

Durante lo sviluppo dell’applicazione, abbiamo usato tante tecnologie messe a disposizione da Android.

### Fragment

Abbiamo utilizzato diversi tipi di fragment nell’applicazione.

Per realizzare i popup di conferma abbiamo utilizzato di Dialog Fragment.

La visualizzazione di dieta e allenamento è invece realizzata attraverso un fragment e un Tab Layout, con tab per ogni giorno della settimana. È stato anche implementato il listener per gestire lo swipe tra le tab.

Per la gestione dei clienti di dietologo e coach abbiamo realizzato due fragment, gestiti come master (contiene la lista dei clienti) e detail (mostra i dettagli del cliente selezionato)

Anche la funzionalità glossario per gli attrezzi utilizza fragment master-detail

### Database

Per salvare i dati dei clienti, dietologi e coach abbiamo realizzato un database SQLite.

Per evitare problemi di concorrenza (thread-safety) abbiamo realizzato la classe SQLiteOpenHelper come singleton, approccio consigliato anche nei libri di testo.

### Service

Le prime versioni dell’app utilizzavano solo il database helper. Questo approccio era poco scalabile, difficile da gestire e pericoloso, perché aumentava il rischio di creare numerose istanze di DB in giro per il codice.

Per risolvere questi problemi abbiamo sviluppato un servizio per l’accesso e uso del database, il DatabaseService. È un servizio di tipo bound che ci permette di scambiare dati e richiamarlo più volte nella stessa sessione.

### aSyncTask

Le query più pesanti, come quella per recuperare le richieste dei clienti di un dietologo, sono state realizzate utilizzando aSyncTask. Queste permettono di eseguire il codice in maniera parallela al thread principale e di aggiornare le view, come la listView che mostra le richieste.

### Notifiche e receiver

Per realizzare le notifiche sui pasti, abbiamo creato un Notification Helper, che crea i canali e le notifiche vere e proprie, come consigliato nella documentazione Android.

Per temporizzare le notifiche abbiamo creato un Alert Receiver, uno speciale Broadcast Receiver che invia le notifiche al momento giusto

### Content Provider e fotocamera

Abbiamo implementato anche la fotocamera nella nostra applicazione in modo che gli utenti possano impostare la loro foto profilo. Il dato viene salvato sul dispositivo tramite content provider, nella cartella tipizzata per le foto. Sono stati gestiti tutti i permessi per accedere alla fotocamera e alla memoria interna del telefono.

### Aggiornamenti meteo REST

Per implementare gli aggiornamenti meteo nell’applicazione abbiamo usato il servizio OpenWeatherMap e la classe Java URLConnection per inviare richieste http. In seguito svolgiamo parsing JSON per estrarre le informazioni dalla risposta http. Per svolgere la richiesta in maniera asincrona al thread principale usiamo una aSyncTask.

### Aprire il tornello

Per aprire il tornello wireless abbiamo usato un microcontrollore ESP-32 e un motorino. Il telefono, alla pressione del pulsante “Apri” invia un messaggio tramite socket che, recepito dell’ESP-32, aprirà il tornello. Quest’operazione è svolta in maniera sincrona su un thread separato da quello principale. Il codice C dell’ESP-32 è inserito nella cartella PlatformIO

# Glossario

## Definizioni

**Coach** –L’esperto che assegna le schede allenamento ai Clienti

**Dietologo** – Il medico che lavora nell’ambito alimentare ed assegna diete ai suoi clienti, dopo una visita dallo stesso.

**Cliente** – La persona che deve seguire una dieta e/o una scheda di allenamento.

**Dieta** – È un piano settimanale degli alimenti da consumare. Possono essere modificati

**Scheda allenamento** – È un piano settimanale degli allenamenti da svolgere in palestra o all’aperto.

**Lista della spesa** – È una lista contenente gli alimenti da comprare e necessari per la dieta.

## Istruzioni operative

Per utilizzare Benessere

-Estrarre la sottocartella “Benessere” dalla cartella “Android Studio” del progetto

-Importare “Benessere” in Android Studio

-Avviare l’emulatore

-Aprire il pannello “file manager” di Android Studio

-In data/data/com.interfacciabili.benessere/databases caricare Benessere.db, che troverete nella cartella principale del progetto

-Avviare nuovamente l’emulatore

PROGETTO TESTATO SU SDK VERSIONE 30 (ANDROID 11)

Account predefiniti

Utente – ID: silvione89 PSW: password

Dietologo – ID: Dietologo1 PSW: password

Coach – ID: Coach1 PSW: password

Attenzione a ID e password, sono case sensitive