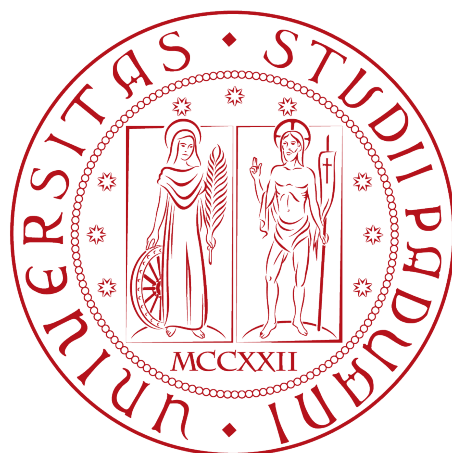


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**Sviluppo di un'app mobile per la gestione dei
pasti aziendali con controllo automatico delle
presenze**

Tesi di laurea

Relatore

Prof. Ombretta Gaggi

Laureando

Erica Cavaliere - 2013450

ANNO ACCADEMICO 2022-2023

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	L'idea	1
1.3	Organizzazione del testo	2
2	Processi e metodologie	3
2.1	Material Design	3
2.2	Processi di sviluppo	4
2.3	Tecnologie	4
2.3.1	Flutter	4
2.3.2	Dart	4
2.3.3	Firebase	4
2.3.4	Figma	5
2.3.5	Android Studio	5
2.3.6	Xcode	5
2.3.7	GitHub	5
2.3.8	Slack	6
3	Analisi dei requisiti	7
3.1	Casi d'uso	7
3.2	Tracciamento dei requisiti	7
4	Progettazione e codifica	8
4.1	Progettazione	8
4.2	Design Pattern utilizzati	8
4.3	Codifica	8
5	Conclusioni	9
	Glossario	10

Elenco delle figure

1.1	Logo dell'azienda RiskApp	1
2.1	Logo del Material Design di Google	4

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 L'azienda

RiskApp S.r.l. (Figura 1.1) è un'azienda con sede a Conselve (PD) che si occupa di sviluppo software per il mondo assicurativo.

È stata fondata nel 2016 e il suo *core business* è lo sviluppo e il mantenimento dell'omonima applicazione, che viene costantemente aggiornata ed estesa per garantire un prodotto che possa rispondere ad ogni esigenza.

Il principale punto di forza di questa piattaforma è quello di stimare le possibili perdite economiche di un'impresa attraverso un algoritmo proprietario che, anche attraverso l'uso dell'intelligenza artificiale, valuta il rischio raccogliendo e combinando una moltitudine di dati da diverse fonti.

Il personale aziendale lavora costantemente per migliorare i propri servizi, ragionando sui possibili problemi che l'utente e l'aziende possono andare incontro, fanno riunioni e call per capire come migliorare e ampliare la piattaforma, tutto svolto in un clima di calma e rispetto tra colleghi.



Figura 1.1: Logo dell'azienda RiskApp

1.2 L'idea

Per poter gestire le spese per i pasti, che preparano in azienda, è stato scelto di sviluppare un'app mobile che permetta di monitorare i versamenti degli utenti, scegliere il piatto del giorno da un menu condiviso e monitorare la [cassa comune](#)^[g].

Deve essere gestita l'autenticazione di ogni utente, dividendo tra utente semplice e utente amministratore e permettere il controllo delle presenze in azienda durante i pranzi.

Ogni utente potrà aggiungere un piatto nel menu, proporre il pasto del giorno, monito-

rare la sua [quota stornata](#)^[g] e la cassa comune, indicare le spese effettuate e modificare i dati personali.

L'amministratore potrà anche gestire le presenze e le spese effettuate dagli stagisti. L'applicazione dovrà essere sviluppata con [Flutter](#)^[g], [Dart](#)^[g] e [Firebase](#)^[g].

1.3 Organizzazione del testo

Il secondo capitolo descrive in che modo è stato creato il prodotto desiderato, quale metodo di sviluppo è stato utilizzato e quali sono le tecnologie adottate per lavorare al progetto.

Il terzo capitolo approfondisce i requisiti con una analisi dettagliata di cosa è stato richiesto.

Il quarto capitolo approfondisce gli strumenti utilizzati per creare il prodotto, la progettazione, i design patter utilizzati e la struttura del codice.

Nel quinto capitolo vengono riportate le valutazioni e le conclusioni personali del prodotto.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Processi e metodologie

Brevissima introduzione al capitolo

2.1 Material Design

Alla base dell'applicazione, è stato scelto di seguire il Material Design (Figura 2.1) (<https://m3.material.io/>) sviluppato da Google, che si concentra su un maggiore uso di layout basati su una griglia, animazioni, transizioni ed effetti di profondità come l'illuminazione e le ombre.

Si tratta di una serie di regole ideate per consentire una buona UX (User Experience) e definire una UI (User Interface) per l'utente da implementare in ambiente Web, Android e in Flutter.

Viene annunciato per la prima volta da Google il 25 giugno del 2014 durante il Google I/O, una conferenza organizzata annualmente da Google a Mountain View, in California.

Venne rinnovato nel 2018 con il Material Design 2, anche chiamato Google Material Theme, introducendo un maggiore utilizzo di angoli arrotondati, spazi bianchi e icone colorate, infine viene rinnovato nel 2021 con il Material Design 3, oppure Material You, introducendo l'uso di tasti più grandi e maggiore uso delle animazioni.

Oggi viene ancora utilizzato il Material Design 3 ed è stato seguito per lo sviluppo dell'app dei pranzi.

Per consentire l'uso dei propri prodotti software a più utenti possibili, il Material Design segue le regole del WCAG (Web Content Accessibility Guidelines) (<https://www.w3.org/WAI/standards-guidelines/wcag/>), mettendo alla base di ogni progetto l'accessibilità, creando così dei prodotti inclusivi, cioè usabili anche da chi ha delle disabilità e deve utilizzare dei software o dei tool appositi, come per esempio un lettore di testo.

I layout devono essere studiati in modo da guidare l'utente nella navigazione della pagina e devono essere dinamici, in modo che le pagine si adattino ad ogni tipo di schermo.

Vengono indicate delle regole precise su come devono essere impostate le componenti, come devono essere raggruppate, lo spazio che deve esserci e tanti altri piccoli ma importanti dettagli che lo sviluppatore deve considerare per permettere all'utente di orientarsi su qualsiasi dispositivo.

Anche Flutter offre una guida sulle componenti che mette a disposizione per lo svi-

luppatore e che sono state ideate per rispettare le regole di Material Design appena descritte (<https://docs.flutter.dev/ui/widgets/material>).

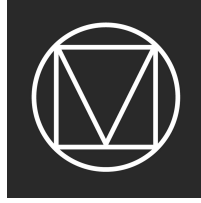


Figura 2.1: Logo del Material Design di Google

2.2 Processi di sviluppo

2.3 Tecnologie

2.3.1 Flutter

Flutter (Figura ??) è un progetto open-source di Google il cui vantaggio principale è la generazione di applicazioni multiplatforma a partire da un unico codice sorgente. Permette quindi allo sviluppatore di concentrarsi sul prodotto da realizzare senza dover preferire un sistema operativo mobile ad un altro.

Per questo motivo è stato scelto di utilizzare Flutter come framework principale, dato che il prodotto finale deve funzionare sia per dispositivi Android sia per dispositivi iOS.

2.3.2 Dart

Il linguaggio sul quale si basa Flutter è Dart (Figura ??), il quale nacque con l'intento di sostituire JavaScript come protagonista delle applicazioni web.

Tra i suoi pregi si elencano il compilatore JIT, migliore gestione della sicurezza, la velocità e la maggiore scalabilità.

Il paradigma principale è l'orientamento agli oggetti, una sua particolarità è data dalla sua attenzione alla null safety per la quale nessun valore può essere nullo a meno che questa possibilità non sia esplicitamente dichiarata.

2.3.3 Firebase

Firebase (Figura ??) è una piattaforma open-source per la creazione di applicazioni per dispositivi mobili e web sviluppata da Google.

Firebase sfrutta l'infrastruttura di Google e il suo cloud per fornire una suite di strumenti per scrivere, analizzare e mantenere applicazioni cross-platform.

Infatti offre funzionalità come analisi, database (usando strutture noSQL), messaggistica e segnalazione di arresti anomali per la gestione di applicazioni web, iOS e Android.

Per lo sviluppo dell'app sono stati utilizzati:

- Firebase Autenticathion, per permettere la registrazione e l'autenticazione di un utente tramite mail e password
- Cloud Firestore, per la gestione del database.

2.3.4 Figma

Figma (Figura ??) è un software per la progettazione di User Interface(UI). Permette infatti di realizzare prototipi delle interfacce, altresì detti mockup, che permettono di illustrare il risultato finale che si desidera ottenere. Questo strumento è stato utilizzato per mostrare e concordare l'interfaccia dell'app al tutor aziendale, prima della fase di codifica.

2.3.5 Android Studio

Android Studio (Figura ??) è un ambiente di sviluppo integrato adibito per la creazione di applicazioni Android e mette a disposizione dei simulatori virtuali di uno o più cellulari con sistema operativo Android.

Il progetto è stato sviluppato interamente con l'uso di questo IDE ed è stato utilizzato il simulatore virtuale di Google Pixel 7 con sistema operativo Android 13 per testare la build dell'app.

2.3.6 Xcode

Xcode (Figura ??) è un ambiente di sviluppo integrato (Integrated development environment, IDE), completamente sviluppato e mantenuto da Apple, contenente una suite di strumenti utili allo sviluppo di software per i sistemi macOS, iOS, iPadOS, watchOS e tvOS.

Per poter testare la build del progetto, è stato utilizzato il simulatore virtuale di iPhone 15 con sistema operativo iOS 17, messo a disposizione da questo software.

2.3.7 GitHub

GitHub (Figura ??) è una piattaforma di hosting per repository git.

Fornisce agli sviluppatori strumenti per migliorare e mantenere il codice come:

- features utilizzabili da linea di comando,
- gestione delle pull request e code review,
- strumenti per l'issue tracking.

La codebase della piattaforma RiskApp è suddivisa in varie repository su GitHub. Per questo progetto, l'azienda ha riservato una repository apposta per permettermi di lavorare in autonomia al codice.

2.3.8 Slack

Slack (Figura ??) è un'applicazione multiplatforma per la messaggistica istantanea tra membri di un gruppo di lavoro.

Una delle funzioni di Slack è la possibilità di organizzare la comunicazione del team attraverso canali specifici, canali che possono essere accessibili a tutto il team o solo ad alcuni membri.

È possibile inoltre comunicare con il team anche attraverso chat individuali private o chat con due o più membri.

Questo software è stato utilizzato per comunicare con il tutor aziendale da remoto e per condividere materiale.

Capitolo 3

Analisi dei requisiti

Breve introduzione al capitolo

3.1 Casi d'uso

3.2 Tracciamento dei requisiti

Capitolo 4

Progettazione e codifica

Breve introduzione al capitolo

4.1 Progettazione

4.2 Design Pattern utilizzati

4.3 Codifica

Capitolo 5

Conclusioni

Glossario

cassa comune viene utilizzato questo termine per indicare i fondi dati dagli operatori aziendali per coprire i pasti. [1](#)

Dart linguaggio di programmazione *open-source* sviluppato da Google. È il linguaggio principale utilizzato per scrivere applicazioni con Flutter. Dart è noto per la sua velocità ed efficienza nella creazione di applicazioni mobili e web. Risulta inoltre staticamente tipizzato, cioè consente una dichiarazione esplicita dei tipi delle variabili e garantisce maggiore robustezza in programmazione. [2](#)

Firestore piattaforma di sviluppo di app mobile di Google che offre una serie di servizi tra cui database in tempo reale, autenticazione utente, hosting di applicazioni e molto altro. È ampiamente utilizzato per la costruzione di app mobile e web in modo rapido e scalabile, grazie alle funzionalità cloud, di notifica e di monitoraggio in real time. [2](#)

Flutter *framework open-source* di Google per lo sviluppo di applicazioni mobile, desktop e webapp utilizzando il linguaggio *Dart*. È basato su widget personalizzabili, puntando su un rapido sviluppo, eccellenti performance, una comunità attiva e supporto per molte piattaforme. [2](#)

quota stornata indica i soldi che il singolo utente deve dare o ricevere dagli altri utenti per i pasti effettuati e le spese sostenute. [2](#)