

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	Metodo di lavoro	4
2.3	Tecnologie	5
2.3.1	Flutter	5
2.3.2	Dart	5
2.3.3	Firebase	5
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	5
3	Analisi dei requisiti	6
3.1	Casi d'uso	6
3.2	Tracciamento dei requisiti	6
4	Progettazione e codifica	7
4.1	Progettazione	7
4.2	Design Pattern utilizzati	7
4.3	Codifica	7
5	Conclusioni	8
	Acronimi e abbreviazioni	9
	Glossario	10
	Bibliografia	11

Elenco delle figure

1.1	Logo dell'azienda RiskApp	1
2.1	Logo del Material Design di Google	3

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 la progettazione, i *design pattern* 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

2.1 Material Design

Alla base dell'applicazione, è stato scelto di seguire il Material Design (Figura 2.1) 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 *User Experience (UX)*^[g] e definire una *User Interface (UI)*^[g] 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.

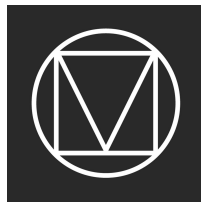


Figura 2.1: Logo del Material Design di Google

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 *Web Content Accessibility Guidelines (WCAG)*^[g], mettendo alla base di ogni progetto l'accessibilità, creando così dei prodotti inclusivi, cioè usabili da tutti i tipi di utenti, anche con disabilità, consentendo a ciascuno un'esperienza fluida e semplice da usare.

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*^[8], 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 sviluppatore e che sono state ideate per rispettare le regole di Material Design appena descritte.

2.2 Metodo di lavoro

Al momento dello stage, RiskApp contava circa dieci dipendenti e ognuno era incaricato di sviluppare e mantenere una parte della loro piattaforma, confrontandosi tra loro ogni giorno per capire come continuare a lavorare.

Il loro metodo di lavoro, si avvicina a un metodo Agile, più precisamente ad uno SCRUM ed è stato il metodo utilizzato anche per lo sviluppo del progetto di stage.

Il Manifesto per lo sviluppo Agile (*Manifesto Agile*. URL: <https://agilemanifesto.org/iso/it/manifesto.html>) è composto da dodici principi fondamentali che descrivono il modo in cui deve lavorare il team, permettendo possibili cambiamenti in corso d'opera e mettendo al primo posto il cliente, rilasciando varie versioni del prodotto funzionante dopo brevi periodi e privilegiando le comunicazioni faccia a faccia.

Lo SCRUM è un *framework* di gestione dei progetti Agile che mira a cinque valori fondamentali e sono impegno, focus, apertura, rispetto e coraggio.

Questo *framework* ha acquisito negli ultimi anni una straordinaria popolarità nel mondo dell'informatica grazie ai vantaggi offerti, come maggiore collaborazione con l'utente finale, il suo contributo al miglioramento continuo e la superiore gestione dei rischi.

L'idea di fondo consiste nel suddividere i periodi di lavoro in *sprint* di durata fissata, caratterizzati da un insieme di obiettivi da realizzare (*sprint backlog*).

Per lo sviluppo del progetto di stage, ogni giorno veniva riportato quanto era stato fatto e veniva mostrato il funzionamento, raccogliendo possibili idee per migliorare o modificare l'app.

Se in corso d'opera venivano incontrate eventuali problematiche sullo sviluppo, si ragionava su come affrontare o modificare il prodotto per risolvere questi problemi, permettendo così di soddisfare ogni esigenza degli utenti finali, in questo caso per soddisfare le esigenze dei dipendenti dell'azienda.

2.3 Technologie

2.3.1 Flutter

2.3.2 Dart

2.3.3 Firebase

2.3.4 Figma

2.3.5 Android Studio

2.3.6 Xcode

2.3.7 GitHub

2.3.8 Slack

Capitolo 3

Analisi dei requisiti

3.1 Casi d'uso

3.2 Tracciamento dei requisiti

Capitolo 4

Progettazione e codifica

4.1 Progettazione

4.2 Design Pattern utilizzati

4.3 Codifica

Capitolo 5

Conclusioni

Acronimi e abbreviazioni

UI [User Interface](#). [3](#), [10](#)

UX [User Experience](#). [3](#), [10](#)

WCAG [Web Content Accessibility Guidelines](#). [3](#), [10](#)

Glossario

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

componenti sono un insieme di *widget* e di elementi che insieme costituiscono un prodotto software. [4](#)

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](#), [10](#)

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. [2-4](#), [10](#)

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](#)

UI indica l'interfaccia grafica che viene utilizzata per le comunicazioni tra uomo e macchina. [9](#)

UX indica l'insieme di sensazioni e ricordi che una persona prova quando si rapporta con un prodotto, cioè tutti gli aspetti che condizionano il prodotto per consentire all'utente di utilizzarlo e capirlo con facilità. [9](#)

WCAG si tratta di una serie di linee guida per l'accessibilità, fornisce una serie di criteri tecnici per rendere siti web, applicazioni e altri contenuti facilmente utilizzabili da tutti i tipi di utente. [9](#)

Bibliografia

Riferimenti bibliografici

Ken Schwaber, Jeff Sutherland. *La Guida Scrum - La Guida Definitiva a Scrum: Le Regole del Gioco*. Novembre 2020.

Siti web consultati

Flutter Material. URL: <https://docs.flutter.dev/ui/widgets/material>.

Manifesto Agile. URL: <https://agilemanifesto.org/iso/it/manifesto.html>
(cit. a p. 4).

Material Design. URL: <https://m3.material.io/>.

WAI Standards Guidelines. URL: <https://www.w3.org/WAI/standards-guidelines/wcag/>.