



## Progetto - “CookIdea”

<b>I - Introduzione</b>	<b>2</b>
I.1 - Descrizione di “CookIdea”	2
I.2 - Obiettivi e Motivazioni	3
I.3 - Funzionalità del Software "cooKIdea"	3
I.3.1 Requisiti necessari del software:	4
I.3.2 Funzionalità facoltative:	4
I.3.3 Descrizione delle ricette	5
I.3.4 Tipologie di Utenti	5
I.4 - Analisi della Concorrenza	6
I.4.1 Giallo zafferano – Sito Web	6
I.4.2 Knoweats!	6
I.4.3 Cookpad	6
I.4.4 Bring	6
I.4.5 Svuota Frigo:	6
I.5 - Cronoprogramma (GANTT)	8
<b>2 - Analisi dei Requisiti</b>	<b>9</b>
2.1. Gestione degli Ingredienti:	9
2.2 Gestione del Menù settimanale	9
2.3 Gestione delle ricette:	10
2.4 Gestione Lista della spesa	10
2. 5 Gestione della funzione di registrazione utente e login	10
<b>3 - Progettazione</b>	<b>11</b>
3.1 - Struttura del Database	11

## V 6.1.1

3.2 - Wireframe, Storyboard con Navigazione	13
3.2.1 Sito Web	13
3.2.2 App Android	15
3.3 Tabella Metodi REST	16
3.4 Materiale utilizzato	18
4 - Sviluppo	19
4.1 Differenze Sito Web e App	19
4.2 - Linguaggi e Tecnologie utilizzati	19
4.3 Architettura del Sistema	20
5 - Risultati ottenuti e Conclusioni	21

# 1 - Introduzione

## 1.1 - Descrizione di “CookIdea”

CookIdea è un applicativo sviluppato per applicazioni Android e siti Web, creato per la gestione e ricerca di ricette di cucina che implementa la generazione automatica di una “lista della spesa” a partire da un Menù ideato dall’utente, sulla base di ricette suggerite dal software.

Di base l'applicativo consente la funzione di ricerca e consultazione delle ricette, l'utente che accede al sito ha la possibilità di **consultare le ricette** messe a disposizione del software tramite quattro modalità:

- ricerca testuale a partire dal nome della ricetta cercata
- ricerca a partire da ingredienti: il sistema restituisce ricette che contengono l'ingrediente ricercato..
- ricerca tramite categorie in cui le ricette sono suddivise per portata (Antipasto, Primo, Secondo, Contorno, Dessert)
- Attraverso una funzione di **ricerca avanzata**, in cui è possibile applicare i seguenti filtri:
  - **Difficoltà**: 3 livelli, rappresentati da Emoji (😄😬😱)
  - **Tempo**: suddividendo i piatti secondo tre *tranches* temporali: 15 min, 15-30 min, sup a 30 min
  - **Portate**: Antipasto, Primo, Secondo, Contorno, Dessert
  - **Provenienza**: regione o paese

## V 6.1.1

La maggior parte delle funzionalità del sistema sono però rese disponibili all'utente dopo il **login**.

L'applicativo infatti prevede la funzionalità di registrazione e login da parte dell'utente, al fine di fornire ulteriori servizi e funzionalità oltre alla mera consultazione delle ricette.

La registrazione da parte dell'utente consente di memorizzare nella sua area personale le proprie ricette preferite.

Lo scopo principale dell'applicativo è facilitare l'organizzazione dei pasti, quindi viene messa a disposizione dell'utente registrato una schermata con il **Menù settimanale** dove inserire il menu per ciascuna giorno della settimana, suddiviso in colazione-pranzo-cena, dove l'utente può inserire una o più ricette o anche degli ingredienti singoli. Utente potrà personalizzare la propria lista andando a cancellare le voci che non gli servono (es. utente cancella pranzo perchè pranza fuori durante la settimana). Gli ingredienti e le ricette inserite nel menù vanno ad alimentare la lista della spesa personale dell'utente:

Con la registrazione l'utente potrà gestire la propria **lista della spesa** attraverso una schermata messa a disposizione dall'applicativo, che potrà integrare e modificare.

## 1.2 - Obiettivi e Motivazioni

Lo scopo principale della nostra applicazione è quello di facilitare la vita degli utenti per quanto riguarda l'organizzazione dei pasti e della spesa settimanale, automatizzando il più possibile queste operazioni, anche nell'ottica di risparmiare e adottare un'alimentazione sana e equilibrata, attraverso la pianificazione dei pasti.

Il sistema è pensato per essere utilizzato da quelle tipologie di utenti che hanno poco tempo a disposizione sia per cucinare che per fare la spesa ma anche per chi vuole scoprire nuove ricette. Si pensa a chi lavora e rincasa tardi o ancora madri e padri di famiglia, o anche agli studenti universitari e giovani lavoratori in cerca di ricette facili e veloci.

## 1.3 - Funzionalità del Software "cooKldea"

### 1.3.1 Requisiti necessari del software:

1. Sistema informativo che comprende un **insieme di ricette** di cucina contenenti le seguenti informazioni:
  - o Ingredienti necessari
  - o procedura di preparazione
  - o provenienza geografica, difficoltà e tempo
  - o immagine illustrativa
2. in base al **Menù settimanale**, che contiene un elenco di ricette dal Lunedì alla Domenica, viene generata una **lista della spesa**
3. genera ricette in base agli ingredienti
4. Funzione di **Login e di Registrazione** con possibilità per l'utente di memorizzare i propri dati e le ricette preferite e di aggiornare questi dati.

è inoltre necessario che l'applicazione sia *responsive* e si adatti alle diverse tipologie di schermo.

### 1.3.2 Funzionalità facoltative:

Seppure non espressamente richiesto in una prima fase di progettazione e pianificazione dell'applicativo si era pensato di dotarlo di alcune funzionalità aggiuntive, elencate di seguito, che per mancanza di tempo non siamo stati in grado di implementare

- sistema di alert tramite Notifiche Push che avvisa l'utente quando non risulta comprato un elemento della lista della spesa
- Profilazione dell'utente in fase di registrazione, raccogliendo informazioni su intolleranze e allergie, preferenze, stili di alimentazione o particolari diete, cibi sgraditi preferiti, capacità culinarie, in base alle quali saranno suggerite all'utente le ricette più consone ai suoi gusti e abitudini.
- Aggiunta di ricette personalizzate da parte dell'utente tramite un template messo a disposizione dell'applicazione
- Aggiunta di ulteriori Categorie con le quali filtrare le ricette, come ad esempio:
  - **Vegetariano**: raccolta di ricette senza carne o pesce
  - **Senza Glutine**: raccolta di ricette che non contengono cereali contenenti glutine e loro derivati
- Funzioni **"Cosa c'è nel Frigo?"**, allo stato attuale è possibile effettuare ricerche a partire dal nome di un ingrediente ma inizialmente si voleva implementare una funzione che a partire da alcuni ingredienti restituisse un elenco di ricette.
- Funzione di filtrare per categorie oppure suggerimenti attraverso le categorie come quella dei "prodotti di stagione"

### **V 6.1.1**

- funzionalità aggiuntive legate alla sicurezza dei dati personali, prevedendo la loro protezione e la sanificazione delle queries per dialogare con il Database, oltre che funzioni di validazione dei campi inseriti dagli utenti.

## **1.3.3 Descrizione delle ricette**

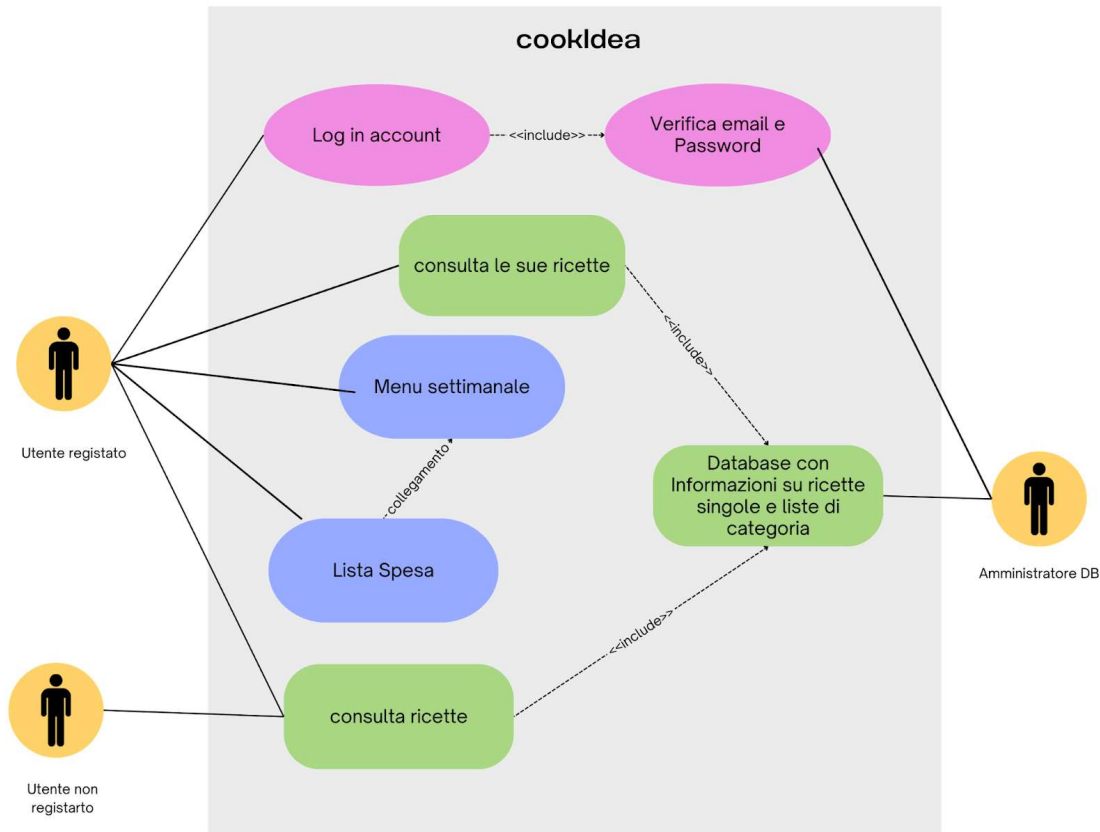
Le schermate di ogni ricetta dovranno contenere le seguenti informazioni:

1. Titolo della Ricetta
2. Foto
3. Informazioni riguardo a: difficoltà, tempo, tipo di portata
4. indica la provenienza geografica
5. Ingredienti e quantità
6. Modalità di preparazione

## **1.3.4 Tipologie di Utenti**

- utente non registrato: può unicamente consultare le ricette presenti nel database
- utente registrato: gestisce ricette personali, lista della spesa e menù della settimana
- amministratore (può modificare il Database)

## V 6.1.1



Le interazioni degli utenti con il sistema sono rappresentate nel diagramma degli Use Case di cui sopra.

## 1.4 - Analisi della Concorrenza

Abbiamo cercato le app mobile e siti web simili alla nostra idea più utilizzati in Italia, allo scopo di prendere spunti sulle funzionalità e grafica di altri software simili.

### 1.4.1 Giallo zafferano – Sito Web

#### Punti Positivi

- layout bello e intuitivo, informazioni e ricette diverse categorie facili da trovare
- Possibilità filtrare le ricette per difficoltà, tempo di preparazione, regime alimentare, ingredienti principali, cottura, altre categorie (es facili e veloci)
- divide lista spesa tra singola ricetta e totale delle ricette [da copiare!]

#### Punti Negativi:

- Troppe categorie diverse nella prima pagina
- Descrizione troppo lunghe
- non è possibile inserire tue liste personalizzare

### 1.4.2 Knoweats!

#### Punti Positivi:

- layout, colori

### **V 6.1.1**

- informazioni nutrizionale ricette

Punti Negativi: poche ricette consultabili

### **1.4.3 Cookpad**

Punti Positivi: app "social" con Ricette aggiunte direttamente da utenti

Punti Negativi

- Solo 2 ingredienti
- Funzione ricerca ricetta per ingredienti limitata

### **1.4.4 Bring**

Punti Positivi

- Grafica e user experience
- Possibilità di condividere lista spesa (tramite whatsapp, Facebook, mail)

Punti Negativi

- obbligatoria la registrazione per poter usare l'App

### **1.4.5 Svuota Frigo:**

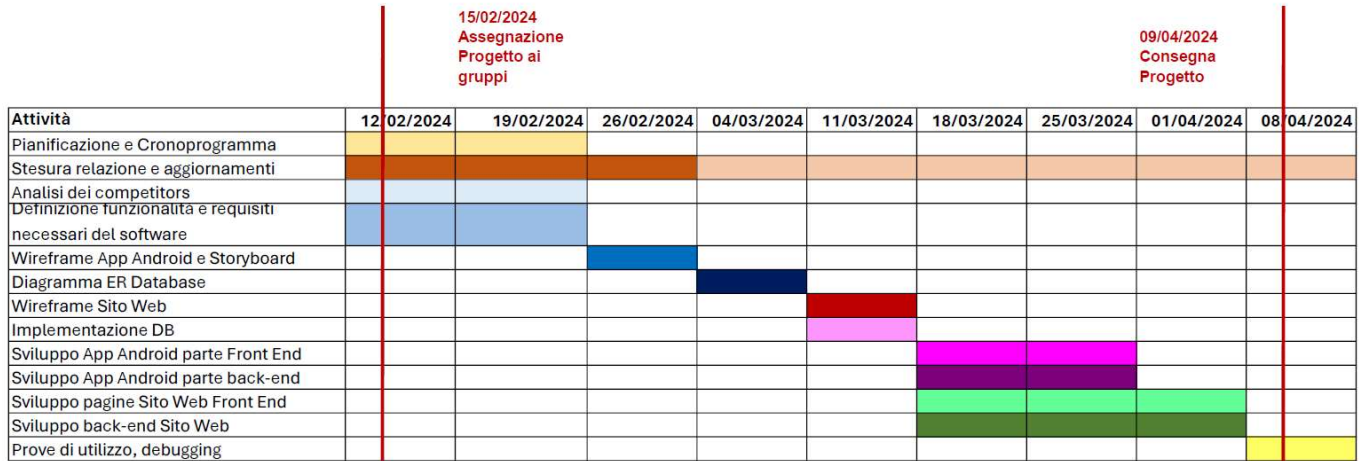
Punti Positivi

- Prende ricette direttamente da siti di ricette
- Funziona senza login
- possibilità di escludere alcuni ingredienti dai risultati di ricerca ricette

Punti Negativi: grafica

### V 6.1.1

Pianificazione temporale delle attività, rappresentate all'interno di un diagramma di Gantt in cui ogni colonna rappresenta le settimane comprese tra l'assegnazione del progetto e la data di consegna prevista.





## 2 - Analisi dei Requisiti

### 2.1. Gestione degli Ingredienti:

L'utente deve essere in grado di visualizzare gli ingredienti per poterli aggiungere alle proprie ricette personalizzate.

Pertanto sono da prevedere le seguenti funzioni:

- **Ricerca e visualizzazione degli ingredienti** tramite menù a tendina, con la possibilità di filtrarli per nome e categoria (es verdure, carne, latticini)
- **Inserimento, modifica ed eliminazione:** l'utente non potrà modificare i dati presenti nel database, ma potrà eliminare o aggiungere gli ingredienti nel contesto della lista della spesa e eliminare o aggiungere ricette alla propria raccolta dei preferiti.
- **Gestione Categorie:** gli ingredienti hanno un attributo relativo alla loro categoria alimentare:
  - frutta
  - verdura
  - panetteria
  - formaggi e latticini
  - carne
  - cereali e derivati
  - spezie e condimento
  - frutta secca e semi
  - legumi
  - Dolci e zuccheri
  - bevanda

### 2.2 Gestione del Menù settimanale

- **Visualizzazione:** all'interno della propria area profilo l'utente può collegare, attraverso un menù a tendina nella pagina della ricetta, le ricette al menù settimanale, valorizzando il campo "giorno" e "tipo di pasto" (Colazione, pranzo, cena, dessert)
- **Modifica il menù:** l'utente ha la facoltà in ogni momento di modificare le scelte di menù.

## 2.3 Gestione delle ricette:

- **Ricerca Base:** con nome ricetta o anche a partire dal nome dell'ingrediente
- **Gestione quantità:** di default le ricette hanno le quantità calibrate per quattro commensali
- **Gestione categorie:** ad ogni ricetta sono associate le seguenti categorie: Provenienza, difficoltà, tipo di portata
- **Aggiunta ricette preferite:** nella sezione personale vengono visualizzate tutte le ricette in cui l'utente ha messo "mi piace" cliccando sull'icona di un cuoricino.
- **Aggiungi a Menù utente:** da singola ricetta utente tramite apposito pulsante può aggiungerla al proprio menù della settimana, specificando giorno e portata

## 2.4 Gestione Lista della spesa

- **Lista della spesa:** nel contesto della lista della spesa collegata a ciascun utente questa è generata automaticamente a partire da un menù settimanale definito dall'utente che a partire delle ricette disponibili le aggiunge alla tabella menù settimanale.

## 2. 5 Gestione della funzione di registrazione utente e login

Sia per il Sito Web che per l'App Android è prevista la funzione di registrazione e Login.

Nella fase di Registrazione l'utente deve compilare i seguenti campi: nome, cognome, password, data di nascita, username (unico), e-mail (unico)

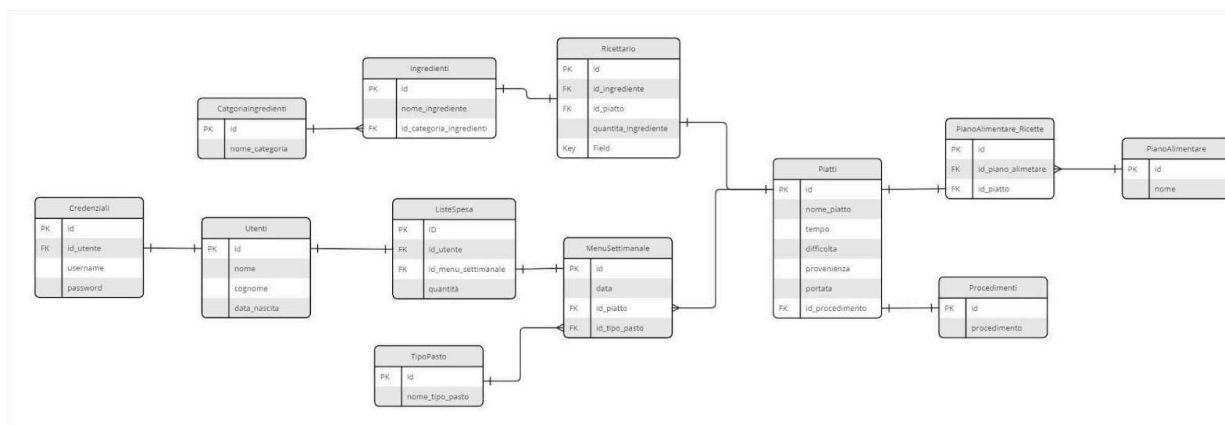
L'utente può modificare i dati relativi a: nome, cognome, data di nascita e password, per quest'ultima è richiesta all'utente di inserire la password vecchia e il sistema verifica la corrispondenza.

## 3 - Progettazione

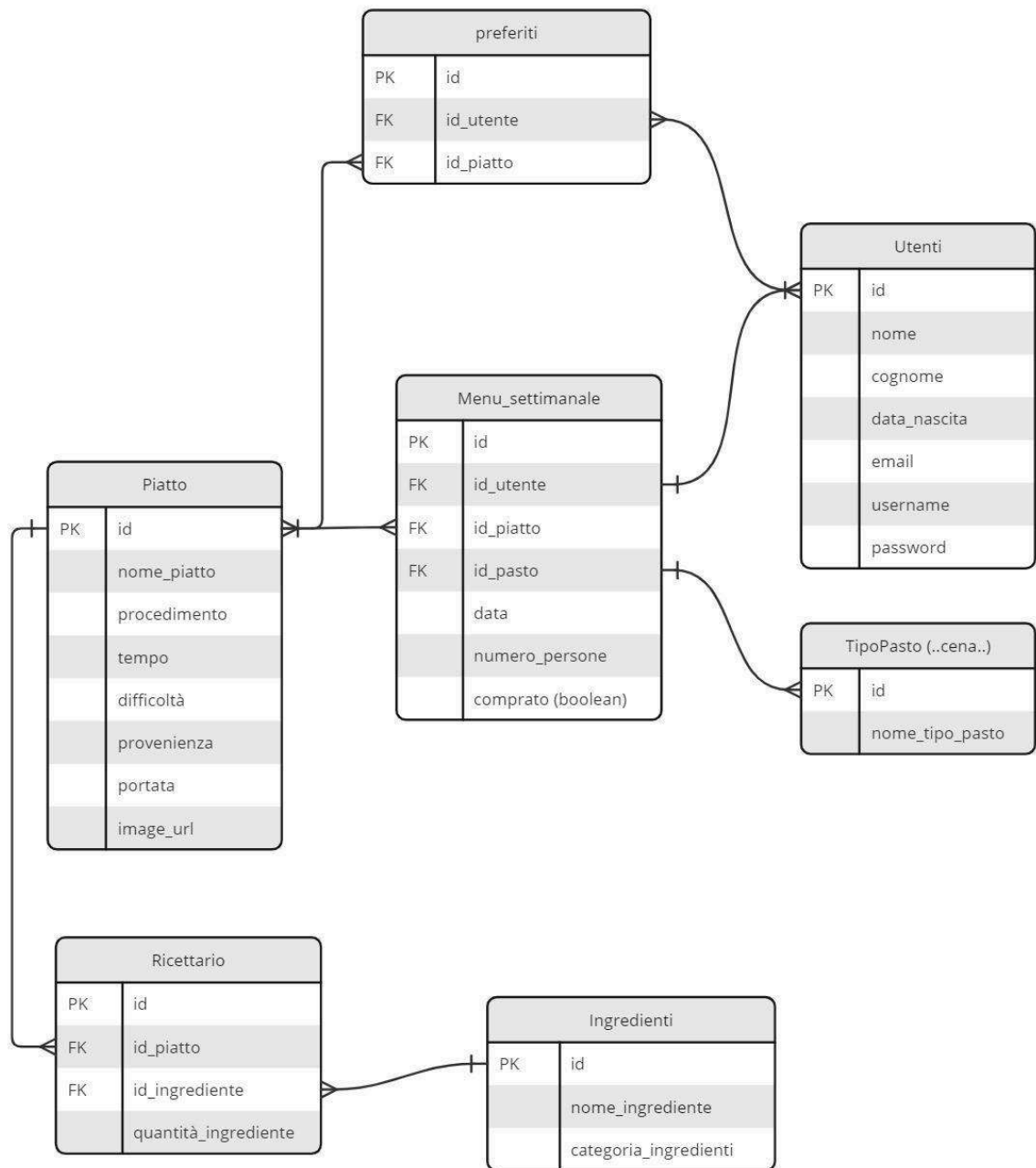
### 3.1 - Struttura del Database

Inizialmente si pensava di utilizzare qualche raccolta di ricette reperibili in rete in modalità Open Source per popolare il nostro Database oppure di recuperare le ricette tramite webscraping ma date le difficoltà dal punto di vista tecnico di recupero di informazioni valide e di gestione di molteplici tabelle e dati, abbiamo deciso di implementare da zero un ricettario inserendo le nostre ricette.

Al **11/03/2024** lo stato dell'arte del Diagramma E-R del Database era il seguente:



Per ragioni di semplificazione si è arrivati a una versione ridotta e denormalizzata:

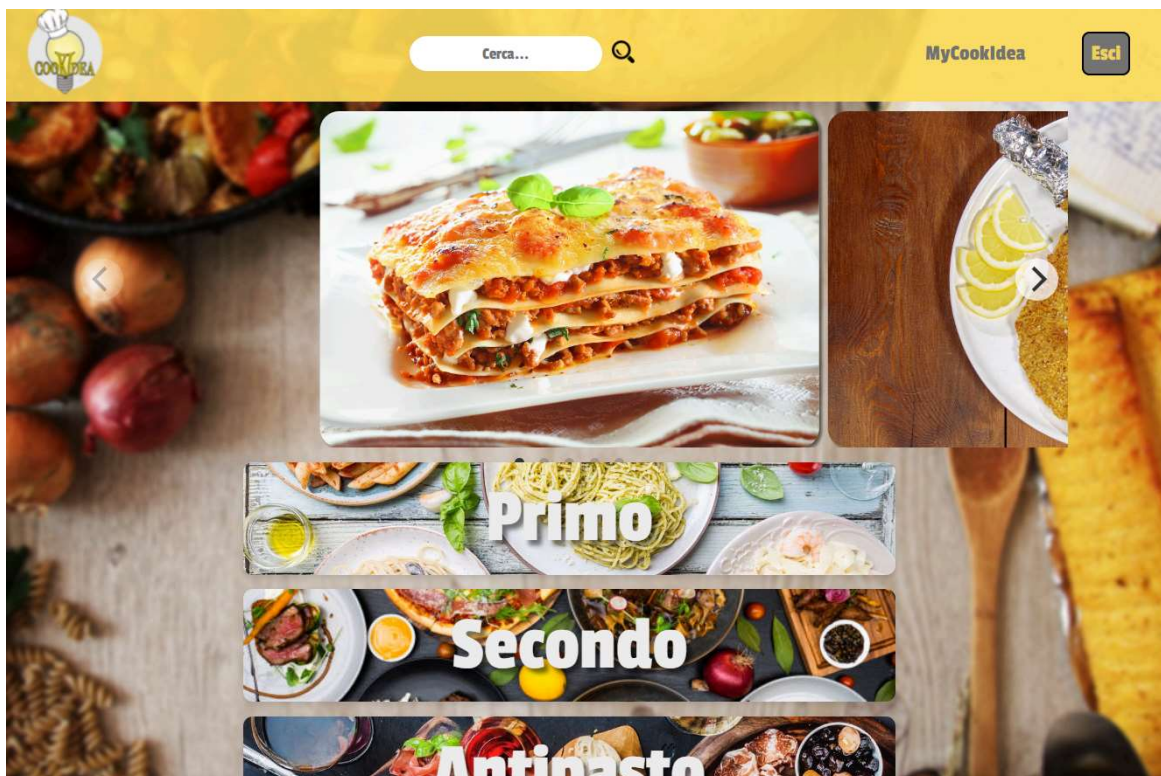


## 3.2 - Wireframe, Storyboard con Navigazione

Già dalle primissime fasi di progettazione si è proceduto all'analisi funzionale del sistema, andando a costruire un insieme di Wireframe con Storyboard. Ogni schermata è corredata da una spiegazione di funzionamento, con l'indicazione delle funzionalità di ogni elemento grafico con cui l'utente può interagire. Sono inoltre indicati tutti i collegamenti tra le varie schermate.

### 3.2.1 Sito Web

La piattaforma consente agli utenti di esplorare una vasta gamma di ricette, suddivise per categorie e tipologie di portate, oltre a offrire uno strumento per la consulta delle preferenze culinarie



- 1) Navigazione intuitiva: Interfaccia user friendly con una navbar ben strutturata per una facile navigazione e consultazione.
- 2) Presenta una barra di ricerca per nome .
- 3) Presentazione delle diverse categorie di ricette tramite schede con immagini di anteprima.
- 4) Gestione dell'utente che permette la registrazione e accesso degli utenti per accedere a funzionalità personalizzate.



Cerca...

MyCookidea 

### Pasta alla Carbonara



Tempo di preparazione  
30


Difficoltà  
2


Portata  
Primo


**Ingredienti**

- Spaghetti :200
- Guanciale :100
- Pepe :5
- Uova :2
- Pecorino :25
- Olio :10
- Sale :5
- Grana :25


**Procedimento**



Cerca...

MyCookidea 

### Pasta alla Carbonara




Tempo:20

Difficoltà:2

Portata: Primo

Provenienza:Lazio

### Risotto ai Funghi Porcini




Tempo:30

Difficoltà:3

Portata: Primo

Provenienza:Piemonte

### Cacio e Pepe



Tempo:15

Difficoltà:2

Portata: Primo

Provenienza:Lazio



## 3.2.2 App Android

Wireframes dell'App Android con descrizione delle funzionalità e dei collegamenti tra le schermate



### 3.3 Tabella Metodi REST

DA FINIRE

ID	Metodo	API / WEB	Url	Endpoint	Parametri	Esempio di JSON	SCOPO
a3	POST	API	cookidea.it	/api/login	{ "username": "mario123", "password": "password123" }	{ "success": true, "user": { "cognome": "Rossi", "data_nascita": "Tue, 15 May 1990 00:00:00 GMT", "email": "mario.rossi@example.com", "nome": "Mario" } }	esegue richiesta login e se esito positivo, restituisce i dati dello user
a4	GET	API	cookidea.it	/api/ricercaPerNome	(nome_piatto) o parte del nome	[{"difficolta": 2, "id": 2, "image_url": "https://staticcookist.akamaized.net/wp-content/uploads/sites/2/2021/10/Ricetta-ai-Funghi-1200x875.jpg", "nome_piatto": "Ricetta ai Funghi Porcini", "portata": "Primo", "procedimento": "Soffriggere cipolla e funghi, aggiungere il riso e sfumare con il vino, cuocere aggiungendo brodo vegetale.", "provenienza": "Emilia-Romagna", "tempo": 30}]	restituisce elenco piatti in base al nome o a una parte del nome
a5	GET	API	cookidea.it	/api/ricercaPerPortata	portata	{ "difficolta": 3, "id": 3, "image_name": "ossobuco.jpg", "nome_piatto": "Ossobuco alla Milanese", "portata": "Primo" }	restituisce elenco piatti in base alla portata
a6	GET	API	cookidea.it	/api/portate		{ "portata": "Primo", "ulteriori": "static/img/primi.jpg" }	restituisce un array / lista di portate prese dalla tabella piatti (piatti portate), senza duplicati
a7	GET	API	cookidea.it	/api/randomPiattoIdNomeImg		{ "id": 5, "image_name": "static/recipes/pizza.jpg", "nome_piatto": "Pizza" }	restituisce 5 piatti random (id, nome, immagine) dal database
a8	GET	API	cookidea.it	/api/ricerca/ricettaFromId	id piatto	{ "difficolta": 2, "id": 1, "image_name": "carbonara.jpg", "nome_piatto": "Pasta alla Carbonara", "portata": "Vino" }	restituisce un piatto/ricetta con tutti i campi di piatto ma con in più con un elenco (lista) di ingredienti in base all'id_piatto
a9	POST	API	cookidea.it	/api/signup	nome, cognome, data_nascita, email, username, password	{ "success": true, "user": { "cognome": "Rossi", "data_nascita": "Tue, 15 May 1990 00:00:00 GMT", "email": "mario.rossi@example.com", "nome": "Mario" } }	esegue richiesta registrazione e, con esito positivo, restituisce i dati del nuovo user inserito
a10	GET	API	cookidea.it	/api/preferitiFromIdUtente	id_utente	{ "id": 2, "nome_piatto": "Ricetta ai Funghi Porcini", "difficolta": 3, "tempo": 30, "procedimento": "Soffriggere cipolla e funghi, aggiungere il riso e sfumare con il vino, cuocere aggiungendo brodo vegetale." }	in base all'id utente, restituisce una lista di piatti preferiti
a11	GET	API	cookidea.it	/api/checkPreferito	id_utente, id_piatto	1	controlla se un piatto è nella lista dei preferiti dell'utente e restituisce 1 in caso positivo e 0 in caso negativo
a12	GET	API	cookidea.it	/api/agg_DatiUtente	id_utente, nome, cognome, data_nascita, password	{ "success": true, "user": { "cognome": "Rossi", "data_nascita": "Tue, 15 May 1990 00:00:00 GMT", "email": "mario.rossi@example.com", "nome": "Mario" } }	aggiornamento dati utente da app Android



ID	Metodo	API / WEB	Url	Endpoint	Parametri	Esempio di JSON	SCOPO
a13	GET	API	<a href="#">cookIdea.it</a>	/api/getUserById	id_utente	{ "success": true, "user": { "cognome": "Rossi", "data_nascita": "Tue, 15 May 1990 00:00:00 GMT", "email": "marco.rossi@example.com", "foto": 4 } }	ottiene uno user in base all'id
a14	GET	API	<a href="#">cookIdea.it</a>	/api/updatePreferito	id_utente, id_piatto		utilizza una logica "switch / toggle" sul button preferito in modo che inserisca o elimini il preferito dal database
a15	GET	API	<a href="#">cookIdea.it</a>	/api/getDailyIngredients	id_utente, data	{ "nome_ingredienti": "Lombata di Vitello", "quantita_ingredienti": 4.0 }	ottiene una lista spesa giornaliera in base alla data e all'utente
a16	GET	API	<a href="#">cookIdea.it</a>	/api/getWeeklyIngredients	id_utente	{ "nome_ingredienti": "Lombata di Vitello", "quantita_ingredienti": 4.0 }	ottiene una lista spesa settimanale in base all'id utente, calcolata sulla data odierna
a17	GET	API	<a href="#">cookIdea.it</a>	/api/getTipoPasto		{ "id": 1, "nome_tipo_pasto": "Colazione" }	restituisce la lista dei tipo_pasto
a18	PUT	API	<a href="#">cookIdea.it</a>	/api/insertWeeklyMenu	id_utente, id_piatto, id_piatto, data		inserisce un piatto nel menu settimanale
a19	GET	API	<a href="#">cookIdea.it</a>	/api/getWeeklyMenu"	id_utente	{ "data": "2024-04-12", "nome_piatto": "Risotto ai Funghi Porcini", "image_name": "risottofunghi.jpg", "nome_tipo_pasto": "Pranzo" }	restituisce un menu settimanale in base all'utente
	GET	WEB	<a href="#">cookIdea.it</a>	/			route del sito web - restituisce la lista portate e 5 immagini random per il carosello
	GET	WEB	<a href="#">cookIdea.it</a>	/connect			come sopra ma verifica se utente loggato, altrimenti riporta a pagina login
	POST	WEB	<a href="#">cookIdea.it</a>	/registrazione	nome, cognome, data_nascita, email, username, password		

ID	Metodo	API / WEB	Url	Endpoint	Parametri	Esempio di JSON	SCOPO
	POST	WEB	<a href="#">cookIdea.it</a>	/login	username, password		
		WEB	<a href="#">cookIdea.it</a>	/logout			
	GET	WEB	<a href="#">cookIdea.it</a>	/web/ricercaPerPortata/<portata>	portata		
	GET	WEB	<a href="#">cookIdea.it</a>	/web/ricerca/ricercaFromId	id_piatto		restituisce ricetta e controlla se è nei preferiti
	GET	WEB	<a href="#">cookIdea.it</a>	/web/ricerca/ricercaPerNome	nome_piatto (o parte del nome)		restituisce elenco piatti in base al nome o a una parte del nome
	GET	WEB	<a href="#">cookIdea.it</a>	/web/updatePreferito	username, id_piatto		utilizza una logica "switch / toggle" sul button preferito in modo che inserisca o elimini il preferito dal database
	GET	WEB	<a href="#">cookIdea.it</a>	/web/favorites	username		restituisce una lista di preferiti in base all'utente
	GET	WEB	<a href="#">cookIdea.it</a>	/web/menu	username		restituisce il menu settimanale dell'utente, diviso per giorno e pasto

## 3.4 Materiale utilizzato

Per realizzare il Logo dell'applicazione, riportato in frontespizio, ci siamo fatti aiutare da una giovane promessa di un Liceo Artistico di Pavia, studentessa del settore artistico-grafico, che lo ha realizzato a partire da un prototipo realizzato su Canva:



Per elementi grafici come icone, font e palette di colori abbiamo utilizzato le seguenti risorse online:

<https://www.flaticon.com/>

<https://colorhunt.co/palettes/pastel>

#808080 (Grigio)

#ffde59 (Giallo)

<https://fonts.google.com/icons>

<https://fonts.google.com/>

- **Alatsi** per i contenuti
- **Passion One** per i titoli

Software online di grafica: Canva, Miro

## 4 - Sviluppo

### 4.1 Differenze Sito Web e App

Il sito Web e l'app sono stati progettati per avere usi diversi in quanto l'app, per via del fatto di essere consultata primariamente da smartphone, risulta essere di uso immediato da parte utente, più comodo e portatile, mentre il sito web risulta lo strumento migliore per la consultazione delle ricette, che si presume saranno consultate dall'utente principalmente su browser e tablet. Per questo motivo l'app inoltre presenta maggiori funzionalità:

- Lista della spesa
- Aggiunta al menù settimanale
- Modifica dati utente e recupero password

Inoltre per non appesantire l'App si è scelto per quest'ultima un design più minimalista, con meno immagini da caricare.

### 4.2 - Linguaggi e Tecnologie utilizzati

Elenco dei linguaggi, tecnologie e framework utilizzati per lo sviluppo:

#### **Android**

Linguaggio di programmazione: Java

Framework: Android Application Framework

Librerie: Retrofit2

API Minima/Massima: API 24/34

Linguaggio Markup Layout: XML

Software per lo sviluppo: Android Studio

WebApp per la progettazione del Layout: Miro

#### **Sito Web**

Linguaggio di markup: HTML, CSS

Linguaggio di stile: CSS

Linguaggio di scripting: JS, Python

**Data ultimo aggiornamento : 09/04/2024**

### **V 6.1.1**

Editor e strumenti di sviluppo: Visual Studio Code, Google Chrome, Postman

Framework: Flask

Framework di Icone per l'interfaccia login e registrazione: Ionicons

### **Backend**

Linguaggio di Scripting: Python

Librerie: Flask

Software per lo Sviluppo: Visual Studio Code, Google Chrome, Postman

### **Database**

Database Server: MySQL Server

Database Management System: MySQL Workbench

WebApp per la progettazione: Miro

### **Strumenti a supporto dello Sviluppo**

Mirò (<https://miro.com/app/>) per la creazione dei Wireframes

Canva (<https://www.canva.com/>) per Diagramma ER

**Strumento di collaborazione:** per il versionamento del codice abbiamo creato un repo in GitHub.

## **4.3 Architettura del Sistema**

Il sistema è basato sul paradigma architetturale del Model-View-Control, in cui un'applicazione viene separata in tre componenti principali che interagiscono tra loro.

**Model:** riguarda i dati e la loro manipolazione. Sono state create le classi modello degli ingredienti, degli utenti, dei pasti, delle ricette e delle porzioni, lato backend ci si appoggia su Python per la gestione, recupero e manipolazione dei dati del database.

#### V 6.1.1

**View:** riguarda la rappresentazione grafica dei dati e dell'interfaccia utente, rappresentati nel sito web dalle pagine HTML e di Layout xml in Android supportati dinamicamente dalla MainActivity e dai Fragment.

**Controller:** agisce da intermediario tra il Model e la View. Gestisce il flusso di dati e l'interazione dell'utente. Nel caso della nostra applicazione questa operazione è gestita tramite il Framework Flask che gestisce le richieste RESTful.

## 5 - Risultati ottenuti e Conclusioni

L'idea originale della nostra applicazione era molto più complessa e prevedeva molte funzionalità aggiuntive che non abbiamo avuto tempo di implementare e che avrebbero reso l'applicazione più completa, con un maggiore grado di personalizzazione da parte dell'utente per quanto riguarda la gestione della menù settimanale, con possibilità di aggiungere singoli ingredienti oltre alle ricette e di aggiungere ingredienti in più alla lista della spesa oltre a quelli generati automaticamente.

In fase di progettazione abbiamo avuto difficoltà nella progettazione di un database che soddisfacesse tutti i requisiti del sistema pertanto abbiamo dovuto semplificare. il diagramma Inoltre durante lo sviluppo abbiamo riscontrato difficoltà per quanto riguarda la funzionalità di Login, di implementazione del Menù e la corretta visualizzazione del Carosello del Sito Web

Nel futuro si ipotizza di poter implementare tutte le funzionalità aggiuntive riportate in Introduzione (paragrafo 1.3.2) e di aggiungere funzionalità legate all'uso dei social sia in ottica di condivisione di ricette personali o preferite tra utenti sia per quanto riguarda l'autenticazione.