RossoPomodoro: documentazione tecnica e progettuale

Sommario

[Analisi dei requisiti 3](#_Toc72835539)

[Requisiti di qualità 3](#_Toc72835540)

[Obbiettivi futuri 3](#_Toc72835541)

[Piano dei metodi e delle tecnologie utilizzate 4](#_Toc72835542)

[l front end 4](#_Toc72835543)

[Il backend 4](#_Toc72835544)

[Specifiche del codice e funzioni 5](#_Toc72835545)

[Specifiche di interfaccia 5](#_Toc72835546)

[Login 5](#_Toc72835547)

[Register 5](#_Toc72835548)

[HomePage 6](#_Toc72835549)

[Ricerca 6](#_Toc72835550)

[Salvati 6](#_Toc72835551)

[Recipe 6](#_Toc72835552)

[Componenti 7](#_Toc72835553)

[CardImage 7](#_Toc72835554)

[Recipebox 7](#_Toc72835555)

[Menù 7](#_Toc72835556)

[(Backend) API pubbliche 8](#_Toc72835557)

[API di registrazione 8](#_Toc72835558)

[Mail API 8](#_Toc72835559)

[API di conferma 8](#_Toc72835560)

[API di Login 8](#_Toc72835561)

[(Backend) Area Personale 9](#_Toc72835562)

[API di esplora ricette 9](#_Toc72835563)

[API di gestione delle ricette 9](#_Toc72835564)

[API di ricerca ricette 9](#_Toc72835565)

[API di gestione delle notizie 9](#_Toc72835566)

[API di gestione del database 10](#_Toc72835567)

[Progettazione del database 10](#_Toc72835568)

[Tabella users 10](#_Toc72835569)

[Tabella ricette 11](#_Toc72835570)

[Tabella salvati 12](#_Toc72835571)

[Tabella lista 12](#_Toc72835572)

[Esperienza, conclusioni e commento personale 13](#_Toc72835573)

# Analisi dei requisiti

Documento ad uso e scopo documentativo per la descrizione del sofware applicazione RossoPomodoro.

RossoPomodoro nasce come un sofware dedito alla scoperta di nuove ricette da parte dell’utente , offre nuove esperienze per scoprire e comprendere le delizie e manicaretti non solo provenienti dalla territorio italiano, ma anche da tutto il mondo.

Il software si rivolge ad utenti che si vogliono approcciare o sono già dediti alla cucina e necessitano di imparare quali siano le migliori tecniche per cucinare. Comprende ricette di tutti i tipi, segnalate dalle opportune difficoltà e per questo consigliate ad un pubblico differente.

L’utente avrà a disposizione una interfaccia grafica user friendly per ricercare facilmente ricette tramite testo o attraverso la propria voce; le ricette saranno poi se possibile salvate e fruibili da egli in un menù dedicato. Le ricette saranno corredate non solo delle informazioni sul procedimento ma anche dei tempi di cottura, della difficoltà, del costo e degli ingredienti.

## Requisiti di qualità

* Il software deve avere un interfaccia user friendly per la facile comprensione delle ricette da parte dell’utente
* Ogni utente deve avere uno spazio privato accedibile in diversa maniera, tramite accesso Google, Facebook oppure accessibile tramite registrazione con email e password
* L’applicazione dovrà supportare le piattaforme di android e ios e dovranno essere supportate per le ultime 5 versioni.
* L’applicazione deve essere disponibile e reperibile pubblicamente sul Play Store e App Galllery
* L’applicazione deve supportare i più recenti standard di sicurezza per la creazione
* I dati devono essere organizzati in strutture sequenziali mettendo in primo piano le informazioni generali ed in secondo piano le informazioni di dettaglio di ogni ricetta (ingredienti, tempi di realizzazione e cottura)

## Obbiettivi futuri

* Il software dovrà supportare un’autenticazione di tipo biometrico per il riconoscimento di volti o impronte digitali
* In caso di mancanza di servizi di rete, la possibilità di consultare le proprie ricette salvate dovrà essere comunque disponibile
* Il software dovrà essere tradotto in lingua inglese per poter rendere disponibile la fruizione dello stesso in tutto il mondo
* Implementazione di riconoscimento di cibo tramite fotocamera del dispositivo e proposta di ricette con cibi al suo interno

# Piano dei metodi e delle tecnologie utilizzate

L’obbiettivo è quello di formare una applicazione fatta, completa e funzionante che descriva una situazione ideale per l’utente: le chiavi per la riuscita di questa ultima è l’uso di tecnologie che permettano una memorizzazione facile e veloce, di modo da garantire all’utente un prodotto con meno difetti possibile.

Per questo si suddivide il lavoro in due parti:

## l front end

Il front end è realizzato in linguaggio javascript, un linguaggio potente per la creazione di applicazioni e interfacce che possono spaziare dall’utilizzo web all’utilizzo come applicazione, l’interfaccia dovrà avere questi criteri:

* Sopportare il cambio di modalità, dalla dark mode (trend diffuso nei cellulari dal momento del lancio da parte di Apple in IOS 13) alla più classica light mode, in grado di soddisfare anche gli utenti più tradizionalisti
* Utilizzare il linguaggio di modo da lavorare in maniera meno distruttiva possibile e strutturare ogni singolo componente, che dovrà ripetersi all’interno della applicazione, in un componente assestante per poter effettuare lavori di manutenzione in modo più agevole
* Utilizzare una repository github per la gestione comoda e veloce delle versioni del sofware di modo da poter recuperare singoli errori man mano ce li si commettono
* Fornire il codice con la documentazione interna adeguata

La tecnologia utilizzata è React, con il framework Ionic, che può garantire un aumento delle prestazioni dato dal Virtual DOM,

La gestione in componenti e rotte rende facile la modifica e la risoluzione di problemi, e mantiene organizzata la gestione dei file e delle istruzioni compilate.

## Il backend

Per il back end si utilizza sempre il linguaggio javascript, per mantenere una concordanza con tutto il lavoro, esso permette una gestione efficace delle rotte e un integrazione perfetta con MongoDB, il database NOSql, oltre a garantire delle prestazioni maggiori e una comprensione maggiore del linguaggio.

Per il database si utilizza una MongoDB, in unione con il framework di accesso Mongoose.

Si utilizza il database MongoDB perché garantisce una sicurezza dei dati presenti all’interno del cluster in unione con una migliore gestione e maggiori performance rispetto ad un classico Database MYSQL.

# Specifiche del codice e funzioni

## Specifiche di interfaccia

L’interfaccia dell’applicazione di suddivide in pagine e oggetti:

* LOGIN
* REGISTER
* HOME
  + HOME
  + SALVATI
    - Liste
    - Ricette
  + RICERCA
* RECIPE

## Login

L’interfaccia Login prevede l’inserimento della mail con cui si è registrato e la password.

In caso un utente non inserisse correttamente le credenziali verrebbe reso a display un popup con un avviso di non inserimento corretto delle funzioni.

Gli errori possono avvenire quando un utente non esiste oppure quando un utente non ha possibilità

## Register

L’interfaccia di registrazione è una semplicissima interfaccia accessibile, il metodo, dopo la corretta registrazione di un utente, richiamerà la API per l’inserimento.

In caso l’utente non inserirà delle informazioni valide, verranno generati degli alert per comunicare all’utente che le proprie informazioni non sono valide.

Se così non fosse il programma procederà con la registrazione e manderà sulla pagina di avvenuta registrazione.

## HomePage

La homepage è suddivisa in tre parti, regolate da un router gestito a sua volta da una barra multifunzione , in questo modo è possibile effettuare switch tra le tre pagine, Home, Ricerca e Salvati, senza necessariamente utilizzare tasti funzione del proprio telefono.

## Ricerca

La pagina di ricerca permette tramite un componente di input la ricerca di un qualsiasi ricetta, all’interno della ricerca è permesso ricerca il cibo che più si aggrada

## Salvati

Nella pagina dei salvati è permesso tramite un componente Segmento, di mostrare due tipi di sezioni, a seconda della selezione della opzione dell’utente:

* In un caso si possono vedere le ricette di un utente, salvate nella navigazione dello stesso, per poter essere facilmente consultabili ed eliminabili
* Nell’altro caso si possono vedere le liste che un utente ha creato

## Recipe

Dopo aver selezionato un a qualsiasi ricetta, si giunge alla pagina recipe, dove vengono mostrati le informazioni delle ricette; viene inoltre messo a disposizione un bottone per poter salvare le ricette all’interno della propria raccolta.

# Componenti

## CardImage

E’ la gestione del componente per la homepage, ossia un componente obreggiato per la visualizzazione delle ricette giornaliere

## Recipebox

E’ la gestione del componente recipe, per la gestione e la visualizzazione delle ricette

## Menù

E’ la gestione dell’avatar e del menù funzionale all’interno delle tre pagine principali di ricerca, permette la disconnessione dal profilo, la visualizzazione delle impostazioni per l’area personale e la visione di notizie pubblicate dall’autore.

# (Backend) API pubbliche

Per la gestione e la conservazione del backend, si hanno diversi metodi, in generale puntati sul mantenimento della sessione, sulla sicurezza e sulla messa a disposizione di servizi all’interno del nostro, la sessione è sestita tramite cookie, per la sessione e il mantenimento della connessione.

All’interno del codice backend abbiamo diverse API:

## API di registrazione

Per la gestione della registrazione l’utente andrà ad inserire le proprie informazioni all’interno del nostro programma, per poi inserire all’interno della API il mio utente andrà a procedere con l’invio della mail e la conferma della registrazione.

## Mail API

Questa API invierà una mail personalizzata, brandizzata per la conferma della registrazione

## API di conferma

L’API di conferma verifica effettivamente l’avvenuta conferma della registrazione e permette l’accesso al portale.

## API di Login

L’API di Login ha la funzione di permettere il login all’interno della nostra applicazione, per poter accedere alla propria area personale. Verrà assegnato quindi un token JWT per il proseguimento della navigazione all’interno della mia area personale

# (Backend) Area Personale

## API di esplora ricette

All’interno della nostra applicazione è possibile esplorare le ricette che vengono caricate giornalmente, costituirà quelle che sono le ricette all’interno della pagina principale, ove è possibile consultarle e salvarle nella raccolta personale

## API di gestione delle ricette

All’interno della API di gestione delle ricette è possibile effettuare una gestione per cui in caso una ricetta sia disponibile sul database, allora la ricetta viene servita, se così non è possibile passarla allora la ricetta viene dedotta e inserita all’interno del database e poi successivamente inserita.

Questa funzione sarà disponibile solo nella beta per poi essere disponibile una totale ricerca all’interno del database nella versione stabile quando e se saranno disponibili abbastanza ricette.

## API di ricerca ricette

Anche in questo caso si è preferito basarsi un metodo di ricerca fondato sul webscraping, per poi reperire informazioni da poter poi inserire all’interno del programma e nel suo database.

## API di gestione delle notizie

Per poter inviare notizie sul cibo, o in questo caso ricette, compresa la poesia di Pablo Neruda si utilizza la Notion API, per poter permettere all’amministratore di sistema di inviare notizie comodamente dal suo profilo Notion.

# API di gestione del database

Per la API di gestione del database si definisce il modello CRUD.

* CREATE
* READ
* UPDATE
* DELETE

Dove si possono inserire, eliminare, modificare informazioni all’interno del database, si possono inserire ricette, liste della spesa, utenti o ricette salvate, e saranno disponibili per gli utenti per poter modificare le proprie informazioni all’interno della propria area personale.

# Progettazione del database

Il progetto prevede una gestione delle informazioni tramite il db mongodb, per la gestione in maniera gerarchica delle informazioni necessarie alla nostra applicazione.

L’applicazione sarà fornita di 4 tabelle

* USERS
* RECIPES
* LIST
* SAVEDS

**Attenzione !! Le tabelle sono sprovviste di ID, che viene puoi aggiunto automaticamente dal software MONGODB.**

## Tabella Users

Come nell’apposito schema ER, abbiamo la tabella sers, in cui è salvato il nome, nome utente , il link della immagine, la mail e l’hash della password, per rendere sicuro l’inserimento della password.

Inoltre la tabella ha due chiavi esterne, una che contiene le ricette salvate e l’altra che contiene le liste della spesa.

const users = new Schema({  
 nome: {type : ***String***, required: true},  
 nickname: {type : ***String***, required: true, unique: true},  
 mail : {type : ***String***, required: true, unique: true},  
 password : {type : ***String***, required: true},  
 flag : {type : ***Boolean***, default: false},  
 id\_lista : { type: ***String***, default: 0},  
 id\_preferiti : { type: ***String***, default: 0}  
});

## Tabella ricette

La tabella ricette è dotata di tutte le informazioni relative alle ricette.

const recipes = new Schema({  
 titolo : {type : ***String***, required: true, unique: true},  
 descrizione : {type : ***String***, default: "non disponibile"},  
 immagine : {type : ***String***, required: true},  
 categorie : {type: ***Array***, required: true},  
 ingredienti : {type : ***Array***, required: true},  
 procedimento : {type : ***Array***, required: true},  
 difficoltà : {type : ***String***, required: true},  
 preparazione : {type : ***String***, required: true},  
 cottura: {type : ***String***, required: true},  
 dosi: {type : ***String***, required: true},  
 costo: {type : ***String***, required: true}  
});

La tabella ricette sarà aggiornata dinamicamente nel momento in cui un utente seleziona una nuova ricetta compatibile.

## Tabella salvati

La tabella salvati conterrà la tabella delle ricette salvate nell’area personale dell’utente

const saved = new Schema({  
 ricette : {type : ***Array***, required: true}  
});

## Tabella lista

const list = new Schema({  
 nome : {type : ***String***, required: true},  
 ingredienti : {type : ***Array***, required: true}  
});

Ultima tabella, collegata alla tabella utente, che predispone una lista della spesa, per l’utente di modo da organizzare gli ingredienti per un eventuale acquisto.

# Esperienza, conclusioni e commento personale

L’esperienza di programmazione predisponeva il mio interfacciamento con due nuove tecnologie, oltre alla progettazione di interfaccia; con questo progetto ho avuto la possibilità di approcciarmi con la programmazione ibrida, con stile di programmazione con Ionic,NodeJS e con il database ad oggetti MongoDB.

Era mia intenzione intraprendere in questo mondo, soprattutto per il mio sogno di diventare uno sviluppatore freelancer e devo dire che durante questa esperienza ho trovato molte difficoltà ma anche molte agevolazioni nella programmazione rispetto all’esperienza di programmazione in PHP e MYSQL

La principale difficoltà è stata cominciare a pensare di utilizzare un framework, come Ionic e la scelta che ho dovuto fare tra Angular, React e Vue; scegliendo React, ho dovuto adattare il mio metodo di programmazione in React, strutturando il mio progetto in componenti ed oggetti.

Questo metodo di programmazione l’ho portato avanti anche nella gestione del mio Backend in NodeJS e nel database, strutturando le rotte a seconda di ogni funzione e suddividendo gerarchicamente.

Ma la difficoltà grande è stata quella di interfacciarsi con i diversi sistemi operativi, su cui andrà ad essere operativa la applicazione: la possibilità di rendere disponibile la applicazione per Ios, facendo fronte

Una delle agevolazioni che ho riscontrato è il facile reperimento, delle funzioni aggiuntive, in quanto npm, il package manager di NodeJS, è molto più immediato nella curva di apprendimento di Composer, il package manager di PHP, ed è molto più completo.

Questo progetto è stato molto formativo per l’acquisizione delle competenze mancanti che mi ero preposto di imparare e ha contribuito molto nella finalizzazione dei miei obbiettivi futuri e nella realizzazione