Immagine che contiene schizzo, clipart, Line art, simbolo

Descrizione generata automaticamente

Università degli studi di Napoli Parthenope

Progetto Ingegneria del Software e

Interazione Uomo Macchina

F1 Universe – SDD

**Roberto Della Corte 0124002666**

**Rocco Del Prete 0124002592**

**Luca Amoroso 0124002512**

Sommario

[Introduzione 3](#_Toc157942616)

[Scopo del sistema 3](#_Toc157942617)

[Obiettivi di progettazione 3](#_Toc157942618)

[Definizioni, acronimi e abbreviazioni 4](#_Toc157942619)

[Riferimenti 5](#_Toc157942620)

[Tecnologie web 5](#_Toc157942621)

[Ambienti di sviluppo 5](#_Toc157942622)

[GitHub 5](#_Toc157942623)

[Adesione agli standard di progettazione PWA 5](#_Toc157942624)

[Sistema corrente 6](#_Toc157942625)

[Sistema proposto 6](#_Toc157942626)

[Decomposizione del sistema 6](#_Toc157942627)

[Hardware/software mapping 6](#_Toc157942628)

[Gestione dei dati persistenti 6](#_Toc157942629)

[Controllo accessi e sicurezza 6](#_Toc157942630)

[Decisioni sul flusso di controllo globale 6](#_Toc157942631)

[Condizioni limite 6](#_Toc157942632)

[Servizi del sottosistema 6](#_Toc157942633)

[Glossario 6](#_Toc157942634)

# Introduzione

## Scopo del sistema

Creare un'esperienza coinvolgente e informativa per gli appassionati di Formula 1 attraverso una Progressive Web App (PWA) che offra aggiornamenti in tempo reale, promuovendo così l'accesso rapido a ultime notizie, statistiche, e momenti salienti del mondo della Formula 1. La PWA mira a fornire una piattaforma centralizzata per gli appassionati, integrando funzionalità quali news dell’ultima ora, accesso diretto ai profili dei piloti e delle squadre, ma anche a calendario e risultati, nonché la possibilità di partecipare a discussioni mediante la sezione dei commenti, creando così una comunità virtuale appassionata e ben informata. L'obiettivo è offrire un'esperienza utente intuitiva e gratificante, sia per i fan casuali che per i seguaci più accaniti della Formula 1.

## Obiettivi di progettazione

Gli obiettivi di progettazione sono:

* **responsive design**, quindi si vuole garantire che l’app sia accessibile da diverse piattaforme e dispositivi, garantendo un’eserienza utente uniforme su desktop, tablet e dispositivi mobili;
* **velocità di caricamento**, cioè ottimizzare le prestazioni per garantire tempi di caricamento rapidi, in modo che gli utenti accedano ai contenuti desiderati nel minor tempo possibile;
* **intuitività**, ossia progettare un’interfaccia utente intuitiva per massimizzare la soddisfazione dell’utente e garantire un’esperienza senza intoppi;
* **sicurezza**, quindi garantire la sicurezza dei dati degli utenti, specialmente considerando la sensibilità delle informazioni personali;
* **persistenza**, cioè capacità della PWA di conservare le informazioni.

## Definizioni, acronimi e abbreviazioni

* **F1** (Formula 1), ossia l’argomento della nostra PWA;
* **PWA** (Progressive Web App), che è un tipo di applicazione web che utilizza tecnologie web per fornire un’esperienza utente simile a quella di un’applicazione nativa;
* **versioning**, ossia il processo di assegnazione e gestione di versioni numeriche ad un codice sorgente durante il suo sviluppo, che permette di tener traccia delle modifiche, permette di consentire a più sviluppatori di lavorare contemporaneamente sullo stesso progetto senza interferenze e permette il ripristino del codice a versioni precedenti nel caso in cui si verifichino problemi con le nuove modifiche;
* **framework**, cioè un’infrastruttura software che fornisce un’architettura predefinita, strumenti e linee guida di progettazione per facilitare lo sviluppo software;
* **microframework**, quindi un tipo di framework che permette di risolvere un problema specifico. Rispetto ai framework sono quindi meno generalisti e più specifici;
* **Flask**, microframework web di Python basato sullo strumento Werkzeug e sul motore di templating Jinja2;
* **Werkzeug**, strumento utilizzato da Flask e che permette di fornire strumenti utili per la gestione e il routing delle richieste HTTP.
* **Jinja2**, che è un motore di templating utilizzato da Flask e progettato per generare dinamicamente documenti, principalmente HTML, utilizzando un linguaggio simile a quello di Python. Viene utilizzato nei framework web, come ad esempio Flask, per generare dinamicamente pagine HTML, includere dati dinamici nei template e semplificare la presentazione dei dati agli utenti;
* **DBMS** (DataBase Management System), ossia un software progettato per creare, gestire e organizzare database;
* **RDBMS** (Relational DataBase Management System), cioè software progettati per creare, gestire e organizzare database relazionali;
* **database relazionali**, ossia database rigorosi che seguono specifici modelli di relazione tra i dati. Le informazioni in questo tipo di database sono organizzate in tabelle con relazioni ben definite tra esse;
* **transazioni**, che rappresentano insiemi di operazioni nell’ambito dei DBMS;
* **atomicità**, che significa che se una parte della transazione fallisce, deve essere annullata l’intera transazione e il database viene ripristinato allo stato precedente, in quanto una transazione viene vista come singola unità indivisibile;
* **coerenza**, che garantisce che una transazione porti il database da uno stato coerente ad un altro, quindi devono essere rispettare le regole di integrità dei dati;
* **isolamento**, che garantisce che l’esecuzione simultanea di più transazioni non interferisca tra loro, in quanto ogni transazione viene eseguita in modo indipendente da tutte le altre;
* **durabilità**, che assicura che una transazione confermata equivale a modifiche permanenti. Ciò significa che i dati saranno immuni da perdite di dati anche in caso di interruzioni di qualsiasi tipo.

## Riferimenti

### Tecnologie web

Per quanto riguarda le tecnologie web abbiamo utilizzato HTML, CSS e JavaScript, per garantire una progettazione moderna e conforme agli standard web, e Python, con il suo microframework Flask, per la gestione del backend, con supporto di Jinja2 per i template e Werkzeug come toolkit web.

### SQLite

Per il mantenimento di dati come utenti, giornalisti e amministratori registrati, ma anche news e commenti, abbiamo utilizzato SQLite, che è un DBMS SQL che ci ha permesso di creare e gestire un database tramite Python.

### Ambienti di sviluppo

Per quanto riguarda lo sviluppo in HTML, CSS e JavaScript abbiamo utilizzato come ambienti WebStorm e Visual Studio Code, mentre per quanto riguarda la parte in Python ci siamo serviti di PyCharm.

### GitHub

GitHub è stato utilizzato come piattaforma di hosting e collaborazione per il versioning del codice sorgente, la gestione delle modifiche e la collaborazione efficace tra i membri del team.

### Adesione agli standard di progettazione PWA

La PWA è stata progettata in modo da aderire alla maggior parte degli standard di progettazione delle PWA, ossia:

* **responsive design**, quindi assicurarsi che l’applicazione sia progettata per essere visualizzata in modo ottimale su una varietà di dispositivi e dimensioni dello schermo, come ad esempio smartphone, tablet e computer;
* **connessione sicura**, cioè utilizzare sempre il protocollo di connessione HTTPS per garantire la sicurezza delle comunicazioni e la protezione dei dati degli utenti;
* **presenza di un manifest**, quindi un file JSON che descrive l’applicazione, inclusi nome, icone, colore di sfondo e altre informazioni che possono essere utilizzate quando l’app viene installata sul dispositivo;
* **presenza di un service worker**, che è un componente JavaScript che funziona in background e consente funzionalità come il caching delle richieste, che permette di migliorare l’affidabilità e le prestazioni dell’app anche in assenza di connessione;
* **installabilità**, in modo da fornire un’esperienza di installazione simile ad una app nativa, che consente agli utenti di aggiungere facilmente l’app alla loro schermata iniziale o alla barra delle applicazioni.

# Sistema corrente

Per confrontarci con i sistemi correnti abbiamo visionato diverse pagine web del settore, in modo da poter prendere spunto, ma anche per migliorare alcune sezioni.

Siamo partiti dal presupposto che confrontarci con il sito web per eccellenza nel settore, ossia quello ufficiale della Formula 1 ([formula1.com](https://www.formula1.com/)), fosse errato, poiché loro sono direttamente associati al marchio Formula 1, mentre il nostro applicativo è più “amatoriale” e diretto ai fan che vogliono consultare notizie nel modo più rapido possibile.

Abbiamo però voluto riprendere diversi aspetti, a livello di interfaccia, del sito ufficiale della Formula 1, poichè crediamo che questo ci metta in posizione di vantaggio rispetto ai nostri diretti concorrenti.

Proprio parlando di concorrenti, portiamo come esempio il sito [Formula Passion](https://www.formulapassion.it/).

Di seguito abbiamo la parte superiore dell’homepage di Formula Passion, compresa di navbar.

Immagine che contiene testo, veicolo, Veicolo terrestre, ruota

Descrizione generata automaticamente

Bisogna innanzitutto dire che questo sito, oltre che di Formula 1, si occupa di motori più o meno a 360°.

Come si può notare, la loro idea è stata quella di mettere in risalto alcune news, dando una dimensione maggiore a questi articoli.

Tramite immagine non si può visualizzare (non si vede la “manina”), ma avevamo il cursore sulla sezione “Calendari” della navbar, che infatti assume una sottolineatura rossa, che tramite click ci porterebbe alla pagina web dedicata appunto alla sezione da noi scelta (in questo caso “Calendari”).

Passiamo ora alla parte successiva dell’homepage, la quale contiene una carrellata di notizie.

Immagine che contiene testo, schermata, Sito Web, Pubblicità online

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Sito Web, Pubblicità online

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Stampa, design

Descrizione generata automaticamente

Infine abbiamo la sezione relativa al footer, ossia la parte finale dell’homepage, che contiene diverse informazioni, come possono essere l’azienda proprietaria del sito e come contattarli, o anche informazioni riguardanti il copyright.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

# Sistema proposto

## Panoramica

Prendendo come spunto l’esempio portato nella sezione precedente, abbiamo pensato che ci fossero diverse cose da poter migliorare.

Bisogna innanzitutto dire che il nostro applicativo ha come finalità principale il mondo della Formula 1, ma potrebbe essere facilmente riadattato in modo da poter contenere al suo interno anche “mondi” collegati, come possono essere la MotoGP, o anche altre competizioni motoristiche.

In generale abbiamo pensato che potesse essere migliorata la chiarezza e l’intuitività sia della navbar che delle varie pagine.

Per quanto riguarda la prima, quella del nostro applicativo vuole essere più minimale e, per le sezioni drivers e teams, si vuole implementare un menù a tendina, che compare al passaggio del cursore sopra alla voce di menù di navigazione (che può essere drivers o teams appunto).

Per quanto riguarda invece le varie pagine, l’obiettivo è migliorarne la leggibilità, visto che nell’esempio precedente, ad esempio, l’homepage è piuttosto dispersiva, con un’infinità di news.

Proprio per questo motivo l’homepage del nostro applicativo avrà solamente le ultime news, e non una lunga carrellata.

Altra aggiunta da implementare è quella di avere sempre la navbar posta nella parte superiore dello schermo, anche quando si scende nella pagina, in modo da poter sempre accedere a qualsiasi sezione dell’applicativo.

Vorremmo inoltre creare delle pagine singole per i piloti (drivers), con informazioni anagrafiche e di carriera, ma anche pagine apposite per le scuderie (teams), con informazioni riguardanti sede ed organigramma, ma anche i risultati complessivi delle stesse in Formula 1.

Si vuole inoltre dare la possibilità di passare dalle pagine dei drivers a quelle dei teams, mediante voci apposite all’interno delle pagine stesse, oltre che dalla solita navbar.

Dalla navbar si può anche accedere al calendario della stagione corrente di Formula 1, tramite la voce schedule, e alle classifiche, tramite la voce standings.

Tutte queste modifiche/aggiunte sono finalizzate a cercare l’informazione richiesta nel modo più veloce possibile, ossia si vuole velocizzare l’interazione tra l’utente ed il sistema, in quanto la soglia di attenzione degli utenti si è abbassata sempre di più negli ultimi anni ed anche perchè abbiamo visto che nei sistemi correnti non sempre basta un unico click per raggiungere l’informazione desiderata.

Per quanto riguarda gli utenti, tutti quelli non registrati potranno accedere a tutte le sezioni dell’applicativo, eccezion fatta per i commenti sotto le news, che rimangono un privilegio per gli utenti registrati

Per quanto riguarda i ruoli di amministratore e giornalista, si vuole implementare un’interfaccia ad hoc per ognuno dei due, in modo da avere tutte le opzioni fornite da quel ruolo in un’unica pagina.

Per rendere l’applicativo più accessibile a tutti, si utilizzerà la lingua inglese per tutte le sezioni, comprese le news.

In definitiva, si vuole dare accesso a quante più notizie sui protagonisti del “mondo” Formula 1, con un’interfaccia che vada incontro alle esigenze di velocità e immediatezza degli utenti, riprendendo lo stile del sito ufficiale della Formula 1 ([formula1.com](https://www.formula1.com/)), ovviamente rapportandolo ad un tipo di applicativo “più piccolo” come il nostro.

## Decomposizione del sistema

Immagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

* **visualizzazione dati**, che si occupa della visualizzazione di tutti i dati, come ad esempio news, schede piloti, schede team, classifiche, calendario, ecc.;
* **interfaccia login/registrazione**, che permette di accedere ad aree riservate, o comunque sblocca delle operazioni aggiuntive;
* **interfaccia utente**, che consente agli utenti loggati di effettuare commenti alle varie news;
* **interfaccia admin**, che permette agli amministratori di fornire credenziali per l’accesso ai giornalisti e consente anche di occuparsi della gestione delle news, con aggiunta, rimozione e modifica delle stesse;
* **interfaccia giornalista**, che consente ai giornalisti di occuparsi della gestione delle news, con aggiunta, rimozione e modifica delle stesse;
* **effettua** **commenti**, che è il componente che permette la gestione dell’operazione omonima;
* **gestione news**, che è il componente che permette l’aggiunta, la rimozione e la modifica delle news;
* **fornisce credenziali**, che è il componente che permette l’aggiunta di un giornalista al database, in modo che questo possa svolgere il suo lavoro;
* **SQLite**, che consente di mantenere su cui si basa l’applicativo.

## Hardware/software mapping

## Gestione dei dati persistenti

Per la gestione dei dati persistenti, come ad es. news, commenti, utenti registrati, giornalisti e amministratori, abbiamo utilizzato SQLite, un DBMS SQL leggero e semplice, sia da gestire che da interrogare. SQLite viene inoltre fornito come libreria incorporabile all’interno delle applicazioni e questo ne semplifica ancor di più il suo utilizzo.

Altri vantaggi riguardano la totale assenza di configurazione o setup complessi ed il supporto delle proprietà ACID (atomicità, coerenza, isolamento e durabilità) che garantiscono l’integrità e la coerenza delle transazioni (insieme di operazioni) in un RDBMS.

## Controllo accessi e sicurezza

## Decisioni sul flusso di controllo globale

## Condizioni limite

# Servizi del sottosistema

# Glossario