Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea in Informatica

Garzone
Il link tra PA, cittadini e commercio locale

Relatore: Prof. Denaro Giovanni

Tutor Aziendale: Fumagalli Aliosha

Relazione di tesi:
Giuseppe Facchi
Matricola 845662

Anno Accademico 2020-2021

A mamma, papà e Gabriele per avermi accompagnato tenendomi per mano durante tutto il mio percorso.

A Ruben e Pietro, fedeli e fondamentali compagni di viaggio per avermi fatto sempre sorridere e avermi aiutato a superare gli ostacoli più difficili.

A tutte le persone, compagni e professori, incontrate in Università che hanno contribuito a farmi amare questa Scienza più di quanto lo facessi già.

Indice

Capitolo 1

Introduzione

Garzone è una soluzione informatica rivolta alla pubblica amministrazione, in particolare ad enti comunali, al servizio dei cittadini e del commercio locale. L'idea è nata dalla necessità di rilancio del commercio locale, oppresso dal progresso dei giganti tecnologici e dalla pandemia. Affiliata all'iniziativa "Soldiarietà Digitale" promossa dal Ministero dell'Innovazione Tecnologica[1], successivamente si è poi evoluta sulla base dei consigli di varie attività commerciali e di amministrazioni pubbliche. Il progetto, dopo la sua pubblicazione, ha inoltre ricevuto il "Premio Top of the Pid - Restart" [2], promosso dalle Camere di Commercio italiane, come miglior modello di business 4.0 per il rilancio del commercio locale. L'iniziativa premia i migliori progetti innovativi che possono agevolare il rilancio dell'economia per uscire dalla profonda crisi provocata della diffusione del Covid-19.

Figura 1.1: Premio Top of the Pid - Restart

1.1 Riepilogo esperienza di stage

Nel mese di Luglio 2020 vengo contattato dall'azienda dglen srl presso Milano. Dopo un colloquio conoscitivo e avermi introdotto le mansioni che mi sarebbero state assegnate, già

dalla settimana successiva comincia il mio periodo di formazione. Il percorso formativo ha previsto inizialmente l'acquisizione di competenze sulla libreria Javascript "React" mediante l'utilizzo della piattaforma Udemy, un repository online di corsi di formazione. Successivamente mi sono state illustrate le modalità di lavoro da adottare e gli strumenti principali da utilizzare durante lo sviluppo del nuovo progetto, come Bitbucket per la condivisione del codice sorgente e di Visual Studio Code per la sua stesura. Nei mesi successivi il compito assegnatomi riguardava lo sviluppo e la manutenzione di Garzone. L'aspetto più significativo e positivo dell'esperienza di stage trascorsa è sicuramente rappresentato dal lavoro di gruppo, senza il quale non sarebbe stato per me possibile superare ostacoli e imparare tutte le metodologie di lavoro con costanza e meticolosità.

1.2 Specifiche progettuali

Figura 1.2: Logo di Garzone

La finalità principale di Garzone riguarda il soddisfacimento della necessità dei commercianti di poter aggiornarsi, tramite un primo approccio, al commercio online. La soluzione, per raggiungere l'obiettivo, deve permettere a un titolare di poter gestire con semplicità il proprio catalogo di prodotti venduti e servizi offerti e poterlo condividere online con la cittadinanza. Inoltre, in un contesto dove la presenza online è strettamente necessaria per far conoscere al pubblico la propria attività, un titolare deve avere a disposizione tutti gli strumenti venire contattato e per far sapere dove svolge la sua attività tramite la costituzione di una vetrina virtuale. Fondamentale è poi l'interazione tra cittadino e commerciante, la quale deve essere garantita mediante l'implementazione di un'interfaccia per scambiarsi messaggi di chat e la corretta gestione degli ordini di ritiro e di consegna a domicilio, eventualmente con pagamento online.
Di enorme rilevanza, per avere una buona accessibilità alla piattaforma da parte dell'utenza, è inoltre previsto l'utilizzo di linee guida grafiche ispirate agli standard pubblicati da AgID (Agenzia per l'Italia digitale)[3]. Gli enti comunali coinvolti dovranno invece avere a disposizione degli strumenti di controllo, per verificare il funzionamento dell'infrastruttura e il suo corretto utilizzo da parte dei commercianti.
Inoltre sarà per loro possibile comunicare con l'utenza mediante l'apposita sezione dell'applicativo a loro dedicato. Infine dovrà essere prevista, tramite l'implentazione dedicata

di una piattaforma, l'amministrazione di tutta l'infrastruttura inizialmente da parte dell'azienda, ma in futuro da terzi.

1.3 Organizzazione del lavoro

Inizialmente lo stage prevedeva la presenza presso la sede aziendale dove il coordinamento avveniva mediante riunioni giornaliere prefissate tra me ed il tutor aziendale, generalmente il mattino e a fine giornata. Nel caso emergessero problematiche, causate in genere dalla mia inesperienza, risultava molto semplice interfacciarsi con i colleghi presenti in ufficio. Con il riaggravarsi della pandemia il lavoro è stato riorganizzato in modalità full-remote tramite l'utilizzo della piattaforma Clickup. Le riunioni venivano svolte giornalmente il mattino e prevedevano in aggiunta un momento dove confrontarsi sui problemi emersi durante l'orario di lavoro. Dopo l'inserimento del collega Agazzi il tutor aziendale ha previsto una suddivisione del lavoro tramite la definizione di task precisi, eseguibili solo dopo una formazione collettiva da lui fornita. A fine giornata era prevista una chiamata di allineamento tra me e il collega per aggiornare l'un l'altro sugli sviluppi effettuati durante la giornata e per chiarire problematiche non risolte singolarmente. Il testing delle implementazioni veniva eseguito dal tutor aziendale, il quale si occupava di segnalare eventuali problematiche e possibili correzioni implementabili.

Capitolo 2

Analisi dei requisiti

Un sistema informatico deve, in genere, risolvere un determinato problema relativo ad una categoria di utenza. Per raggiungere l'obiettivo deve quindi fornire un certo numero di funzionalità relative alla gestione delle informazioni, possedendo caratteristiche precise di efficienza e affidabilità. I requisiti riguardano per l'appunto la comprensione e la descrizione del problema da risolvere mediante l'implementazione di funzionalità e di qualità desiderate per il sistema.

2.1 Stakeholders

Si definisce "stakeholder" una qualsiasi entità dotata di comportamento che interagisce con il sistema informatico in questione. Quest'ultimo è inoltre considerato anch'esso uno stakeholder quando interagisce con altri sistemi informatici, nel caso del progetto Garzone quando interagisce, ad esempio, con il sistema di pagamento adottato.

2.1.1 Attori primari

Un'entità che utilizza direttamente la soluzione per perseguire i suoi obiettivi è considerata un attore primario. In garzone gli attori primari sono rappresentati da:

Negozi Entità costituite da un punto vendita fisico in un determinato comune registrato sulla piattaforma. Sono rappresentati da un titolare o referente che ha il compito di gestire il proprio catalogo e interagire con i clienti finale utilizzando le varie funzionalità

Comuni Organi della pubblica amministrazione che delegano un funzionario facente parte della giunta eletta per la gestione e il controllo dei negozi registrati alla piattaforma.

Il loro ruolo è meramente di supervisione, possono infatti abilitare o disabilitare la visualizzazione di negozi nel marketplace. Inoltre hanno la possibilità di scambiare comunicati con i negozi.

Amministratore Persona o gruppo di persone organizzate per la gestione e la supervisione complessiva della piattaforma. L'amministrazione avviene mediante l'interazione con negozi, comuni e clienti che necessitano di supporto o di interventi particolari, come la disabilitazione per condotte scorrette. In Garzone questa mansione è per ora affidata all'azienda stessa, ma in futuro è prevista una delegazione a terzi.

2.1.2 Attori finali

Sono definiti attori finali coloro che usufruiscono del servizio messo a disposizione per raggiungere i propri obiettivi. In Garzone è previsto un solo attore finale: il cliente.

Cliente I clienti rappresentano il bacino di utenza della piattaforma e hanno la possibilità di interagire con i negozi per raggiungere i propri obiettivi, quali l'acquisto di beni/servizi, la comunicazione diretta con i corrispondenti referenti, la richiesta di preventivi, la prenotazione di appuntamenti, la visualizzazione dei cataloghi e altro.

2.1.3 Attori di supporto

Generalmente gli attori di supporto offrono alla soluzione servizi esterni necessari per il suo corretto funzionamento. Spesso sono sistemi informatici, ma possono anche essere rappresentati da società/organizzazioni o persone fisiche

Servizio di autorizzazione dei pagamenti In Garzone è prevista un'interfaccia ad un servizio di autorizzazione dei pagamenti per autorizzare i pagamenti correttamente ed evitare tentativi di frode. Inoltre ha il compito di registrare le transazioni dirette tra Clienti e Negozi.

Servizio di autenticazione degli utenti Per la corretta gestione degli utenti e la prevenzione di attacchi informatici sui dati di registrazione, Garzone si interfaccia a un servizio esterno di registrazione e autenticazione degli utenti. Esso ha il compito di comunicare con il sistema informatico e garantire il corretto flusso di registrazione, accesso e gestione delle utenze.

Servizio di geolocalizzazione Per offrire un'esperienza utente migliore, in Garzone è prevista l'implementazione di un'interfaccia ad un servizio di geolocalizzazione, il quale ha il compito di interpretare indirizzi geografici forniti dal sistema e restituire delle coordinate geografiche precise.

2.2 Requisiti

Un requisito è una capacità o una condizione a cui il sistema, e più in generale, il progetto, deve essere conforme[4]. La modellazione dei requisiti avviene secondo le necessità degli utenti del sistema per risolvere le proprie problematiche.

2.2.1 Requisiti funzionali e non funzionali

I requisiti funzionali riguardano, attraverso l'implementazione delle varie funzionalità, il comportamento del sistema informatico nel suo complesso. Tra i requisiti funzionali rientrano anche i casi d'uso. Vengono definiti invece requisiti non funzionali tutti i requisiti che non sono determinanti nel funzionamento della soluzione informatica (Garzone), ma sono relativi alle sue proprietà nel suo complesso, come ad esempio prestazioni, scalabilità, usalibilità e prestazioni.

2.3 Casi d'uso

Un caso d'uso, seguendo le linee guida RUP (Rational Unified Process), è un insieme di istanze di casi d'uso, in cui ciascuna istanza è una sequenza di azioni che un sistema esegue per produrre un risultato osservabile e di valore per uno specifico attore[5]. Per convenzione, i casi d'uso che prevedono la parola chiave "gestisci" riuniscono gli obiettivi separati CRUD (create, retrieve, update, delete) in un unico caso CRUD.

2.3.1 Casi d'uso in Garzone

I casi d'uso possono essere descritti, in genere, in diversi modi. Per la definizione dei casi d'uso relativi a Garzone, si è optato per una stesura in formato breve, ovvero riepiloghi brevi e precisi relativi ai soli scenari di successo. Di seguito vengono elencati i principali casi d'uso.

Registra Negozio Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, compila manualmente i dati relativi alla sua registrazione. Il sistema, dopo la

convalida dei dati e mediante l'interfaccia con il servizio di autenticazione degli utenti e di geolocalizzazione, registra l'utenza.

Registra Cliente Il cliente compila manualmente i dati relativi alla sua registrazione. Il sistema, dopo la convalida dei dati e mediante l'interfaccia con il servizio di autenticazione degli utenti e di geolocalizzazione, registra l'utenza.

Gestisci Vetrina Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, aggiunge o modifica prodotti e servizi del suo catalogo. L'utente in questione inserisce manualmente i dettagli del prodotto o servizio. Il sistema convalida i dati e registra l'inserimento o la modifica. I clienti saranno in grado di navigare sulla vetrina del negozio e visionarne il catalogo.

Gestisci Promozione Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, aggiunge una promozione relativa a un prodotto/servizio offerto. L'utente in questione inserisce manualmente i dati relativi alla promozione tra cui la data di scadenza. Il sistema convalida e registra la promozione. I clienti saranno in grado di visionare solo le promozioni attive del negozio ed eventualmente usufruirne.

Gestisci Appuntamento Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, crea un evento per un determinato orario referenziando un cliente coinvolto. L'addetto che registra l'appuntamento inserisce i dati manualmente e, dopo convalida e registrazione, il sistema notifica il cliente della creazione dell'evento.

Crea Ordine Il cliente, dopo essersi autenticato, dalla pagina di vetrina del negozio aggiunge prodotti messi a disposizione dal negozio al carrello. Il sistema dopo averne calcolato il totale richiede al cliente, tramite interfaccia grafica, l'inserimento manuale di dati relativi all'evasione dell'ordine. Nel caso in cui il cliente volesse che il suo ordine sia evaso ritirandolo in negozio non sarà necessario specificare un indirizzo di consegna, viceversa dovrà essere inserito. Il sistema, dopo la conferma da parte del cliente, registra l'ordine e il negozio viene notificato dell'avvenuta creazione.

Accetta Ordine Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, accetta la richiesta di ordine effettuata dal cliente. Dopo l'accettazione dell'ordine, il negozio procede con la sua evasione, specificando per gli ordini di consegna a domicilio un eventuale costo di consegna. Il sistema registra il cambio di stato dell'ordine. Il cliente dopo la modifica viene notificato automaticamente dell'avvenuto aggiornamento e del totale aggiornato, comprensivo del costo di consegna inserito.

Paga Ordine Il cliente, dopo l'autenticazione e dopo l'approvazione dell'ordine che prevede un pagamento online da parte del negozio e l'eventuale inserimento del costo di consegna per gli ordini di consegna a domicilio, procede con il pagamento online. Il sistema calcola il totale dell'ordine comprensivo del costo di consegna e delega al sistema di autorizzazione dei pagamenti il completamento dell'operazione. Il sistema notifica il negozio e il cliente dell'avvenuto pagamento e aggiorna lo stato attuale dell'ordine.

Evadi Ordine Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, modifica lo stato dell'ordine. Per gli ordini con pagamento online il negozio procede con l'effettiva consegna e il cambio di stato dell'ordine. Per gli ordini che prevedono il ritiro in negozio la preparazione dei prodotti/servizi richiesti per il ritiro e il successivo cambio di stato dell'ordine. Il sistema registra il cambio di stato dell'ordine e notifica il cliente.

Invia Preventivo Il negozio, tramite il suo titolare o l'addetto all'utilizzo della piattaforma, inserisce manualmente i dati relativi a un preventivo. Il sistema, dopo aver convalidato i dati inseriti, invia il preventivo contenente i dettagli di un possibile ordine al cliente. Il cliente accetta il preventivo e il sistema aggiorna lo stato dell'ordine notificando il negozio dell'avvenuto aggiornamento.

Accetta Preventivo Il cliente, dopo la sua autenticazione e la ricezione di un preventivo da parte del negozio, accetta la proposta. Il sistema aggiorna lo stato dell'ordine e notifica il negozio dell'avvenuto cambiamento.

Invia Messaggio Il cliente, dopo la sua autenticazione, avvia una conversazione di chat con il negozio (o viceversa). Dopo la validazione del messaggio, il sistema provvede alla sua registrazione. Il negozio (il cliente) viene notificato della sua ricezione e viene abilitato ad inviare a sua volta un messaggio.

Abilita Negozio Il comune, dopo la sua autenticazione, abilita o disabilita un negozio, rendendone impossibile l'accesso e la visualizzazione sulla piattaforma.

Gestisci Comunicazione Il comune, dopo la sua autenticazione, aggiunge una comunicazione, la quale sarà visibile ai negozianti e agli utenti del marketplace.

Gestisci Comune L'amministratore di sistema, dopo la sua autenticazione, provvede a gestire i dati relativi ad un comune. Dopo la compilazione manuale dei dati del comune, il sistema li convalida e registra/aggiorna l'istanza.

Gestisci Negozio L'amministratore di sistema, dopo la sua autenticazione, provvede a gestire i dati relativi ad un negozio. Dopo la compilazione manuale dei dati del negozio, il sistema li convalida e registra/aggiorna l'istanza.

Gestisci Cliente L'amministratore di sistema, dopo la sua autenticazione, provvede a gestire i dati relativi ad un cliente. Dopo la compilazione manuale dei dati del cliente, il sistema li convalida e registra/aggiorna l'istanza.

Caso d'uso	**Attori coinvolti**
Registra Negozio	Negozio, Amministratore, Servizio autenticazione, Servizio di geolocalizzazione
Registra Cliente	Cliente, Servizio autenticazione, Servizio di geolocalizzazione
Gestisci Vetrina	Negozio, Amministratore
Gestisci Promozione	Negozio, Amministratore
Gestisci Appuntamento	Negozio, Cliente, Amministratore
Crea Ordine	Cliente
Accetta Ordine	Negozio, Amministratore
Paga Ordine	Cliente, Servizio autorizzazione pagamenti
Evadi Ordine	Negozio, Amministratore
Invia Preventivo	Negozio
Accetta Preventivo	Cliente
Invia Messaggio	Negozio, Cliente
Abilita Negozio	Comune, Amministratore
Gestisci Comune	Comune, Amministratore, Servizio autenticazione, Servizio di geolocalizzazione
Gestisci Negozio	Negozio, Amministratore, Servizio autenticazione, Servizio di geolocalizzazione
Gestisci Cliente	Cliente, Amministratore, Servizio autenticazione, Servizio di geolocalizzazione
Gestisci Comunicazione	Comune, Amministratore

Tabella 2.1: Riepilogo Casi d'uso e Attori coinvolti

2.4 Modellazione astratta e struttura del progetto

Garzone ha lo scopo di rendere possibile la transizione vera e propria di un negozio fisico al digitale fornendo ai negozianti una piattaforma gestionale efficiente e ai cittadini un efficacie canale di comunicazione. Per ogni attore è prevista la realizzazione di una propria piattaforma con un dominio web dedicato.
Tutte le piattaforme comunicano per la gestione dati con un sistema backend unico,

il quale a sua volta si interfaccia con sistemi per l'autorizzazione dei pagamenti, per l'autenticazione degli utenti e una base di dati.

2.4.1 Pannello Negozio

Il pannello del negoziante prevede l'implementazione delle seguenti funzionalità:

- **Registrazione**: consiste in una sezione dove viene richiesto al negoziante di registrarsi, selezionando prima il marketplace (comune) in cui è ubicato e poi inserendo manualmente sia dati personali che dati fiscali relativi al negozio fisico
- **Accesso**: prevede una sezione dove un negozio registrato effettua l'accesso previa una registrazione già effettuata e l'effettiva abilitazione da parte del comune
- **Dashboard**: contiene una data visualization tramite grafici dei dati più rilevanti riguardo la gestione del negozio
- **Gestione dati relativi all'utenza registrata**: prevede una sezione in cui è possibile per il negoziante modificare dati fiscali relativi al negozio o dati personali
- **Gestione funzionalità attive**: offre la possibilità al negoziante di abilitare/disabilitare funzionalità di Garzone relative al suo negozio, come ad esempio i pagamenti online
- **Gestione prodotti/servizi**: da questa sezione è possibile gestire mediante creazione, modifica ed eliminazione tutti i prodotti/servizi presenti nel catalogo del negozio fisico
- **Gestione promozioni**: da questa sezione è possibile inserire, modificare ed eliminare promozioni
- **Gestione appuntamenti**: da questa sezione è possibile inserire, modificare ed eliminare appuntamenti
- **Gestione ordini**: da questa sezione è possibile gestire gli ordini ricevuti dai clienti, i quali seguiranno un flusso di evasione prestabilito
- **Invio messaggi**: tramite questa sezione il negoziante potrà comunicare solo con clienti hanno precedentemente inviato un messaggio al negozio
- **Visualizzazione comunicazioni**: contiene le comunicazioni create dal comune in cui il negozio è registrato

2.4.2 Pannello Comune

Il pannello del comune prevede l'implementazione delle seguenti funzionalità:

- **Accesso**: prevede una sezione dove un comune registrato effettua l'accesso, previa una registrazione già effettuata dall'amministratore di Garzone e l'effettiva abilitazione
- **Dashboard**: contiene una data visualization tramite grafici dei dati più rilevanti riguardo la gestione del negozio
- **Gestione dati relativi all'utenza registrata**: prevede una sezione in cui è possibile per il comune modificare dati fiscali relativi all'organo amminsitrativo e dati relativi al suo referente
- **Abiltazione negozi**: da questa sezione è possibile abilitare o disabilitare i negozi del proprio comune (marketplace)
- **Gestione comunicazioni**: tramite questa sezione il comune potrà comunicare ai negozianti del proprio marketplace avvisi rilevanti

2.4.3 Pannello Amministratore

Il pannello di amministrazione prevede l'implementazione delle seguenti funzionalità:

- **Accesso**: prevede una sezione dove l'amministratore può accedere alla sua piattaforma
- **Dashboard**: contiene una data visualization tramite grafici dei dati più rilevanti riguardo negozi, comuni e clienti
- **Gestione comuni**: tramite questa sezione l'amministratore può gestire, inserendo, modificando o eliminando, i dati dei comuni
- **Gestione negozi**: da questa sezione è possibile gestire tramite inserimento, modifica ed eliminazione, i dati fiscali, i dati relativi al negozio fisico, le chat, gli ordini e gli appuntamenti dei negozianti nei vari marketplaces
- **Gestione clienti**: tramite questa sezione l'amministratore può gestire, inserendo, modificando o eliminando, i dati dei clienti e relativi ordini, chat e appuntamenti
- **Gestione promozioni**: tramite questa sezione l'amministratore può gestire, inserendo, modificando o eliminando, i dati delle promozioni create dai negozianti

2.4.4 Pannello Cliente

Il pannello destinato ai clienti, invece prevede lo sviluppo delle seguenti funzionalità:

- **Registrazione**: consiste in una sezione dove viene richiesto al cliente di registrarsi, inserendo manualmente i suoi dati personali
- **Accesso**: prevede una sezione dove il cliente può accedere alla piattaforma di Garzone
- **Visualizzazione marketplace**: prevede la possibilità per il cliente di visualizzare tutti i negozi, registrati e abilitati dalla piattaforma, di un comune da lui selezionato
- **Visualizzazione promozioni**: revede la possibilità per il cliente di visualizzare tutte le promozioni inserite dai negozi e attualmente attive (non scadute) per il marketplace selezionato
- **Visualizzazione negozio**: prevede la possibilità per il cliente di visualizzare i dettagli di registrazione, il catalogo prodotti/servizi con la possibilità di aggiunta al carrello per un successivo ordine, le promozioni e gli slot di appuntamento già occupati di un negozio specifico selezionato dal marketplace
- **Invio messaggi**: tramite questa sezione il cliente, previa registrazione e accesso, ha la possibilità di inviare messaggi ad un negozio selezionato
- **Richiesta preventivo**: tramite questa funzionalità un cliente può richiedere un preventivo al negoziante indicando le sue necessità
- **Gestione ordini**: da questa sezione è possibile gestire gli ordini inviati ai negozi, i quali seguiranno un flusso di evasione prestabilito
- **Gestione appuntamenti**: da questa sezione è possibile gestire, modificando o annullando, gli appuntamenti presi con i negozi dei vari marketplaces
- **Visualizzazione comunicazioni**: contiene le comunicazioni create dal comune (marketplace) selezionato

I domini della soluzione costituiscono una struttura gerarchica ad albero, in cui il pannello cliente è consultabile tramite dominio principale, mentre gli altri pannelli tramite sottodomini. L'idea generale segue questa struttura:

Figura 2.1: Struttura domini e sottodomini Garzone

Le varie piattaforme comunicano tra loro interfacciandosi al servizio di autenticazione delle utenze e ad un sistema backend condiviso, il quale a sua volta si interfaccia a un'apposita base di dati e al servizio di autorizzazione dei pagamenti.

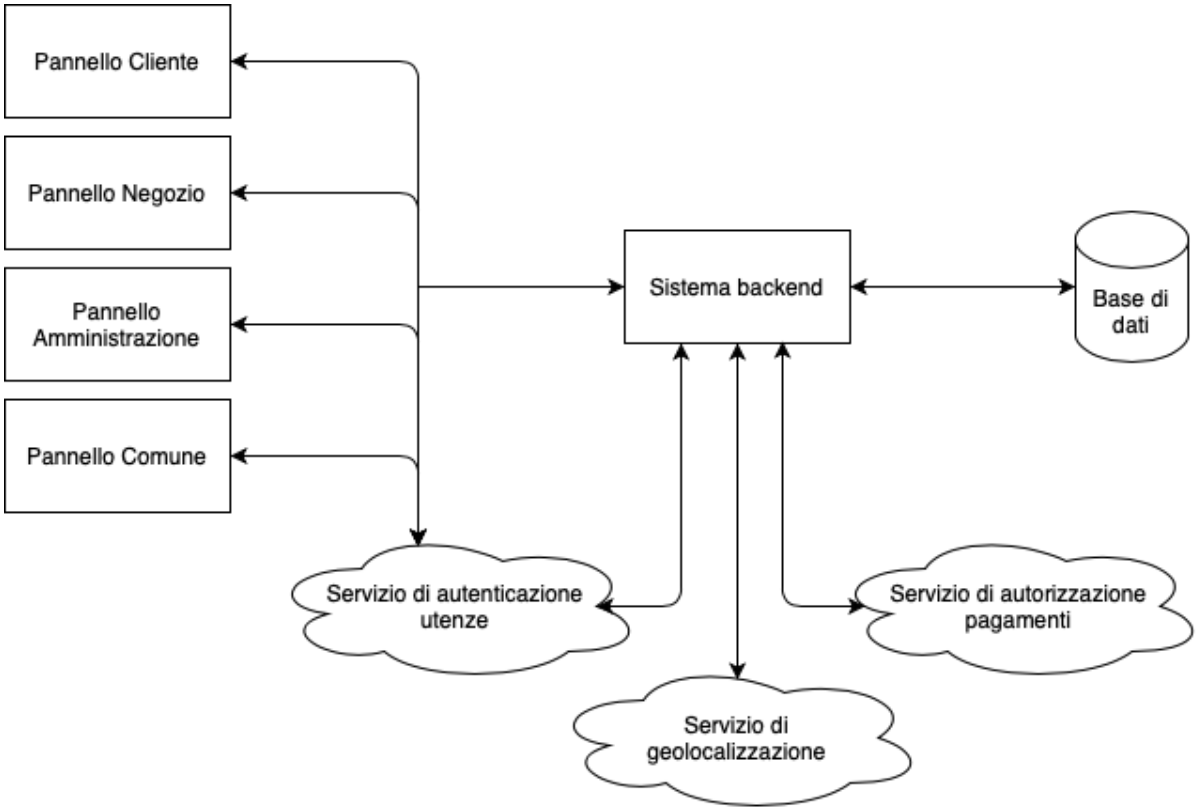

Figura 2.2: Modello complessivo astratto di implementazione

Capitolo 3

Componente tecnologica

3.1 Introduzione

La realizzazione del progetto prevede, dopo la definizione di una precisa architettura, l'utilizzo di diverse componenti tecnologiche, come ad esempio l'utilizzo di un framework, per soddisfare i vari requisiti utente.

3.2 Strumenti utilizzati per l'organizzazione del lavoro

3.2.1 ClickUp

Ai fini di organizzare per il meglio il lavoro da svolgere, l'utilizzo della piattaforma ClickUp ha permesso al team di sviluppo la suddivisione dei compiti da svolgere mediante la definizione di macro-task, organizzati in dashboard visibili dal settore amministrativo e tecnico. Un macro-task rappresenta l'implementazione di un caso d'uso utente e corrisponde ad un rilascio in produzione. Inoltre prevede un'univocità all'interno della dashboard ed è visibile e modificabile dal reparto di amministrazione per supervisionare l'andamento del progetto. A sua volta un macro-task è suddiviso in task, i quali sono organizzati nei diversi pannelli (admin, negozio, utente, comune) riprendendo il codice univoco a cui è associato il macro-task. Ogni task è inserito nella dashboard gestita dal team di sviluppo e corrisponde all'implementazione di una micro-funzionalità. Deve soddisfare precisi requisiti, di seguito elencati.

- **Durata Limitata**: un task deve prevedere un tempo di implementazione da una alle quattro ore complessive
- **Atomicità**: un task deve prevedere l'implementazione/correzione di funzionalità singole

- **Univocità**: un task non deve essere mai ripetuto, per implementare bugfixes è necessario creare un nuovo macro-task e di seguito organizzarne la suddivisione
- **Reperibilità**: un task deve essere facilmente trovato ricercando parole chiave definite nel titolo, il quale segue lo standard «*COD # - Titolo breve*»

Ogni task nel flusso di lavoro può assumere diversi status:

- **Open**: il task è appena stato creato e non è ancora stato implementato
- **In Progress**: il task è in corso di implementazione da parte del team di sviluppo
- **Review**: il task è in corso di revisione da parte dei preposti al testing
- **Closed**: il task è concluso e testato

Un macro-task può passare nello status "closed" solo quando tutti i task a lui relativi sono in closed. A questo punto è possibile procedere con il rilascio della piattaforma in produzione.

Figura 3.1: Esempio di organizzazione macrotask - tasks di una funzionalità di Garzone

3.2.2 Bitbucket

Per consentire la condivisione ddel codice sorgente tra il team di sviluppo è stato previsto l'utilizzo della piattaforma Bitbucket, un servizio di hosting per progetti implementati tramite utilizzo del protocollo di version control Git. Per l'implemntazione del progetto sono state create quattro apposite repository:

- **garzone-serverless**: la quale comprende tutto il codice sorgente relativo alla componente backend
- **garzone-webapp-admin**: relativo al pannello admin
- **garzone-webapp-negozio**: relativo al pannello negozio
- **garzone-webapp-comune**: relativo al pannello comune
- **garzone-webapp-utente**: relativo al pannello utente

Le operazioni di version control sono state effettuate tramite l'utilizzo di Github Desktop, un'applicazione che utilizza il protocollo Git mediante un'interfaccia grafica. Ad ogni task del progetto implementato è correlato un commit creando quindi un riferimento al flusso di lavoro definito in partenza.

3.2.3 Separazione ambienti di lavoro

La separazione degli ambienti di lavoro è necessaria per permettere al team di sviluppo di lavorare alle implementazioni di test parallelamente a quelle in produzione. In questo modo si consente un corretto utilizzo della piattaforma da parte dell'utenza su una versione testata e la possibilità da parte degli sviluppatori di procedere con il live deploy solo dopo aver testato il deploy di test. Al progetto vero e proprio firebase-hosted ne è stato quindi affiancato un altro parallelo, il quale presenta lo stesso identificativo *(piattaforma-garzone)* con l'aggiunta del suffisso *"-test"*. I due progetti presentano la medesima suddivisione di sottodomini e si interfacciano a backend differenti, i quali a loro volta si interfacciano a due istanze di basi di dati differenti. Per l'avvio e la sua messa in produzione, il progetto prevede l'utilizzo di script appositi per ogni pannello, ognuno dei quali ha una funzione specifica di avvio o di deploy. Di seguito vengono proposti gli script relativi alla parte di backend e relativi alla componente frontend del pannello utente, seguiti da una panoramica generale dei servizi a cui si interfacciano le varie piattaforme.

Pannello Utente

yarn start:dev	• Autenticazione su *piattaforma-garzone-test* • Funziona nella directory *garzone-webapp-utente*
yarn start:prod	• Autenticazione su *piattaforma-garzone* • Funziona nella directory *garzone-webapp-utente*
yarn deploy:dev	• Pubblica l'applicazione nel progetto Firebase *piattaforma-garzone-test* nel multi-site di default • Funziona nella directory *garzone-webapp-utente*
yarn deploy:prod	• Pubblica l'applicazione nel progetto Firebase *piattaforma-garzone* nel multi-site di default • Funziona nella directory *garzone-webapp-utente*

Figura 3.2: Documentazione degli script di avvio/deploy del pannello utente

Garzone Serverless

npm run-script serve:dev	• Avvia l'emulatore in locale collegandosi mediante proxy all'istanza db **DEV** • Funziona nella directory *garzone-serverless/**functions***
npm run-script serve:prod	• Avvia l'emulatore in locale collegandosi mediante proxy all'istanza db **PROD** • Funziona nella directory *garzone-serverless/**functions***
npm run-script deploy:dev	• Pubblica le functions nel progetto Firebase *piattaforma-garzone-test* • Funziona nella directory *garzone-serverless/**functions***
npm run-script deploy:prod	• Pubblica le functions nel progetto Firebase *piattaforma-garzone* • Funziona nella directory *garzone-serverless/**functions***

Figura 3.3: Documentazione degli script di avvio/deploy della componente backend (firebase functions)

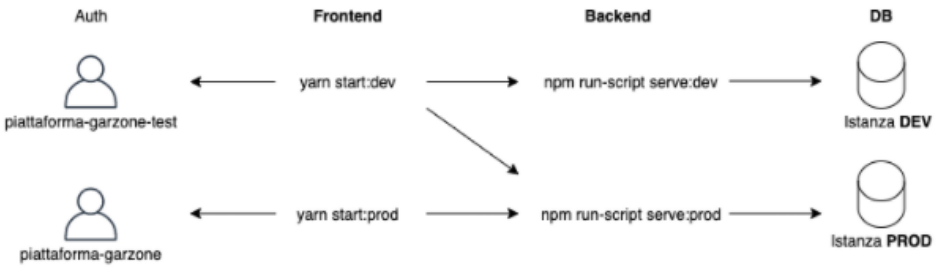

Figura 3.4: Sommario script di avvio dei progetti di test e live e relative interfacce

3.3 Javascript e NodeJS

Il progetto è interamente realizzato tramite l'utilizzo del linguaggio Javascript. Fu introdotto nel 1995 come strumento per inserire programmi nelle pagine web del browser Netscape Navigator. Successivamente è stato via via adottato da tutti gli altri browser grafici, rendendo realizzabili moderne applicazioni Web con le quali si può interagire direttamente, senza dover necessariamente ricaricare la pagina. Vie anche utilizzato in siti più tradizionali per offrire varie forme di interattività.[6]. Node.js è stato creato tramite l'utilizzo di javascript. In particolare si tratta di un ambiente asincrono ed event driven per la costruzione di applicazioni distribuite basate su Javascript. Node.js non è un framework, lavora infatti a livelli più bassi, è sostanzialmente un interprete con alcune librerie di base non native del linguaggio Javascript, ad esempio per l'I/O su file, per l'implementazione di server per vari protocolli standard. Sfrutta il loop di gestione degli eventi di Javascript, in particolare sulla possibilità di definire funzioni callback per implementare uno schema di I/O non bloccante.

Linee guida stesura del codice Il team di sviluppo per la stesura del codice ha adottato uno standard comune a tutte le sue componenti. La struttura dei file di codice sorgente è stati quindi organizzata in:

- **Screens**: ogni schermata dell'applicazione è rappresentata da un file di codice sorgente in questa directory
- **Components**: ogni componente complesso all'interno di una schermata deve essere inserito in un file di codice sorgente apposito, per consentire una maggiore leggibilità e manutenibilità della soluzione
- **Controllers**: ogni componente dell'UI per interfacciarsi con la logica dell'applicativo comunica con un controller che ha lo scopo di delegare questo compito alla parte di backend
- **Assets**: ogni componente grafica o file di dati statico è organizzato in questa cartella e utilizzato in vari punti del codice
- **Config**: i file di configurazione delle librerie esterne, la validazione dei dati e impostazioni varie, quali palette di colori e stringhe statiche, sono organizzati in questa cartella
- **Utility**: tutte le funzioni delegate a compiti complessi, come il calcolo del totale dei prezzi degli articoli di un carrello, sono contenute in file in questa directory

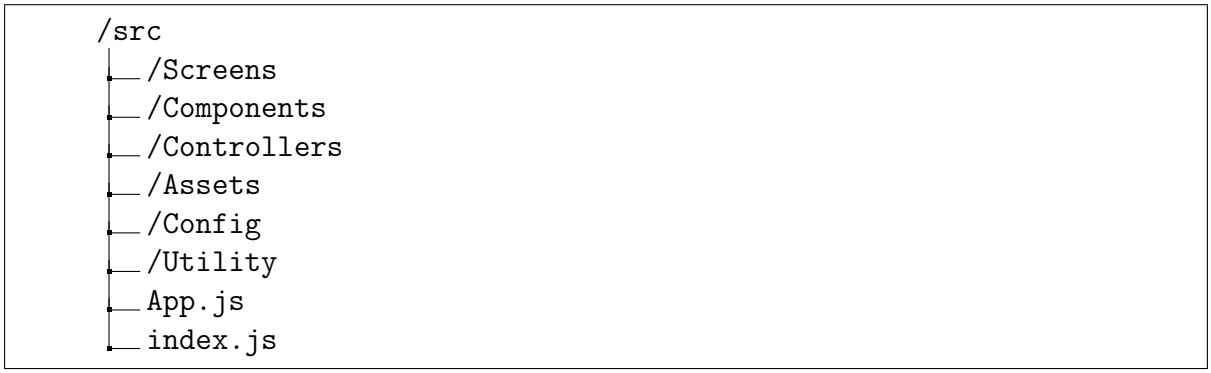

Figura 3.5: Struttura delle directory e dei file di un pannello di Garzone

- *App.js*: rappresenta il codice sorgente radice dell'interfaccia grafica dell'applicazione react
- *index.js*: rappresenta il codice sorgente radice dell'applicazione stessa e ha il compito di eseguire l'inizializzazione delle componenti logiche e grafiche, come lo Store di Redux, una libreria di React per la gestione dello stato globale

Inoltre è stato previsto uno stile di stesura ispirato a Airbnb JavaScript[7], che contiene un elenco di best practices per evitare errori di scrittura e con lo scopo di coordinare il metodo di stesura di codice del team di sviluppo.

3.3.1 NPM e Yarn

NPM è un sistema di gestione pacchetti scritti in Javascript in ambiente Node.js. Di norma contengono collezioni di funzioni alcune delle quali esportate, ovvero "visibili" all'esterno del modulo e quindi richiamabili dal progetto. Per la realizzazione di Garzone sono stati utilizzati numerosi pacchetti, con il compito di interfacciarsi a API esterne e realizzare componenti grafiche complesse come ad esempio mappe interattive. Inoltre è stato utilizzato per l'implementazione di script di esecuzione e rilascio dell'applicativo con configurazioni ad hoc. Per le componenti frontend è stato invece utilizzato Yarn, un packet manager costruito a un livello di astrazione superiore di npm, fornendone una semplificazione della sintassi delle istruzioni eseguibili.

3.4 Componente Frontend

3.4.1 Linee guida grafiche per l'interfaccia utente

3.4.2 ReactJS e antd

3.4.3 Less

3.5 Componente Backend

3.5.1 Google Cloud

Cloud SQL

Firebase

Firebase Functions

Realtime Database

3.5.2 AWS

Simple Email Service

3.5.3 Stripe

Stripe Connect

Capitolo 4

Funzionalità e relative implementazioni

4.1 Modello di Dominio e Diagramma ER

4.2 Gestione utenze

4.2.1 Registrazione

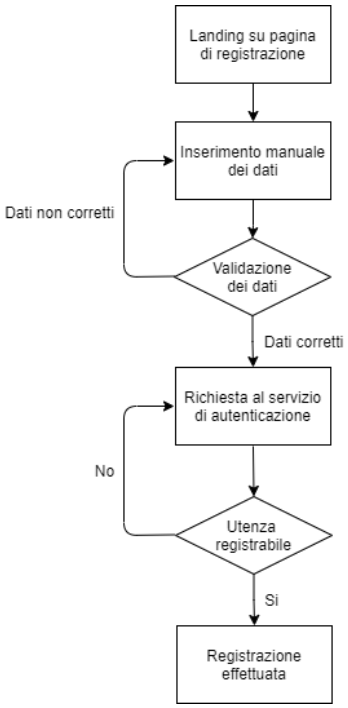

Figura 4.1: Flusso di registrazione del negozio

4.2.2 Accesso

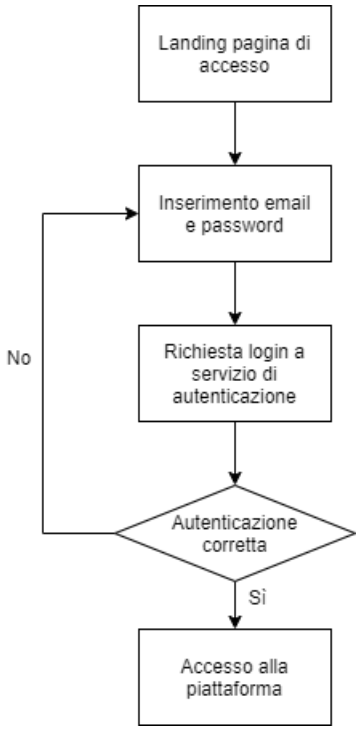

Figura 4.2: Flusso di login del negozio

4.3 Catalogo Prodotti e Catalogo Servizi

4.3.1 Gestione immagini

4.4 Promozioni

4.4.1 Moment.js

4.5 Ordini

4.5.1 Preventivi

4.6 Chat

4.7 Appuntamenti

Capitolo 5

Sicurezza degli applicativi

5.1 Consistenza del dato

5.1.1 Validazione Front-end e Back-end

5.2 Whitelist chiamate API

5.2.1 HTTPS

Conclusioni

5.3 Dati sull'utilizzo

Bibliografia

[1] Ministero per l'innovazione tecnologica e la transizione digitale. *Solidarietà Digitale al servizio di studenti e commercianti*. `https://solidarietadigitale.agid.gov.it/iniziative/`, 2020.

[2] UNIONCAMERE. *Covid: 7 idee innovative per la ripartenza Premiati a Maker Faire i vincitori di Top of the Pid*. `https://www.puntoimpresadigitale.camcom.it/paginainterna/premio-top-of-the-pid`, 2020.

[3] Team per la trasformazione digitale, Ministero per l'innovazione tecnologica e la transizione digitale. *Linee guida di design per i servizi web della PA*. `https://docs.italia.it/italia/design/lg-design-servizi-web/it/bozza/index.html`, 2020.

[4] Jacobson I., Booch G., Rumbaugh J. *The Unified Software Development Process*. Addison-Wesley, 1999.

[5] Rational Software. *Rational Software White Paper*. IBM, 18880 Homestead Road, Cupertino, CA 95014, tp026b, rev 11/01 edition, 1998.

[6] Marijn Haverbeke. *Eloquent Javascript*. Editore unico Hoepli Milano, via Hoepli 5, 20121 Milano, 2 edition, 2015.

[7] airbnb. *Airbnb JavaScript Style Guide()*. `https://github.com/airbnb/javascript`, 2012.