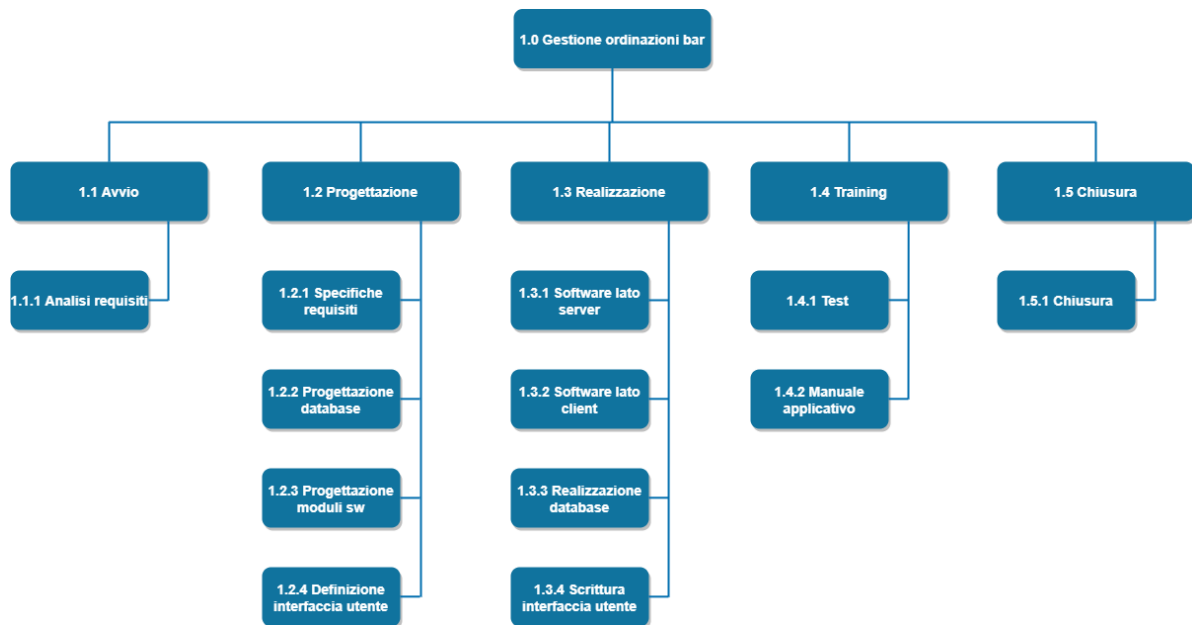


Progetto gestione ordinazioni bar

1. Attività del progetto

1.1 WBS



1.2 Stima tempi

Progetto:	Gestione ordinazioni bar	Codice progetto:	T00
Data:	30/04/2021	Revisione	0
Cliente:	Bar ISII	Sponsor	ISII
WBS	Descrizione	Durata(gg)	
1.1 AVVIO			
1.1.1	Analisi requisiti		1
1.2 PROGETTAZIONE			
1.2.1	Specifiche requisiti		1
1.2.2	Progettazione database		1
1.2.3	Progettazione moduli software		1
1.2.4	Definizione interfaccia utente		1
1.3 REALIZZAZIONE			
1.3.1	Software lato server		5
1.3.2	Software lato client		5
1.3.3	Realizzazione database		2
1.3.4	Scrittura interfaccia utente		2
1.4 TRAINING			
1.4.1	Test		1
1.4.2	Manuale applicativo		1
1.5 CHIUSURA			
1.5.1	Chiusura		1

2. Analisi dei requisiti

2.1 AS IS : Com'è adesso?

Ogni persona della classe deve raccogliere tutte le prenotazioni degli altri compagni. Deve "convertirli" nei vari codici del bar (Es. 2 Pepsi = 2P). Questo può portare a errori e non è immediato farlo ogni volta. Una volta raccolti tutte le prenotazioni la persona addetta dovrà mandare il messaggio su WhatsApp al bar. Il bar molto probabilmente, leggendo i messaggi, si segnerà su carta i vari prodotti da preparare.

2.2 TO BE : Cosa vogliamo ottenere?

Obiettivo del progetto: rendere l'ordinazione più user friendly. Ogni persona ordinerà per proprio conto attraverso Telegram, potrà modificare l'ordine senza essere vincolato.

Il bar avrà un client dove vedrà tutte le ordinazioni. Lì potrà fare un check su quelle già preparate, in modo da avere tutto in modo organizzato. Avendo tante informazioni nel database (le ordinazioni di ogni giorno) fare delle varie statistiche in questo modo si potrà vedere quali sono i prodotti più consumati.

2.2.0 Obiettivi generali

Nelle fasce orarie di non-attività verrà fatto un backup dei dati: avremo una tabella che salverà tutti gli ordini in modo da fare successivamente delle analisi dei dati.

2.2.1 Obiettivi fisici

Avere un computer su cui gira il client nel bar, qui verranno visualizzate tutte le ordinazioni.

3. Use case

3.1 Utente

- LOGIN: l'utente si dovrà loggare la prima volta con la password della classe.
- ORDINARE: l'utente può creare il suo ordine e confermarlo

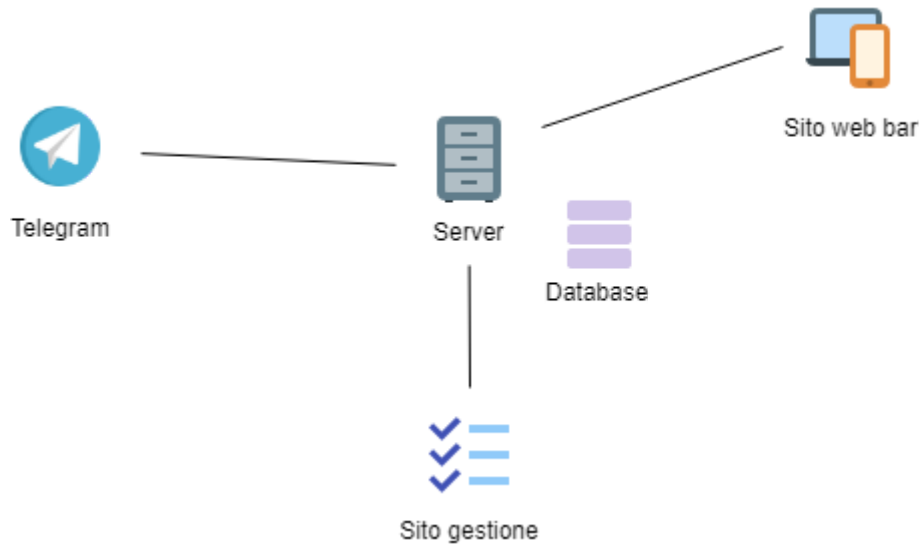
3.2 Utente amministratore

- LOGIN: l'utente si dovrà loggare per poter accedere al sito gestionale.
- MODIFICA: l'utente può fare le varie modifiche sul database

3.3 Omino bar

- LETTURA: l'utente potrà solo visualizzare gli ordini.

4. Schema fisico



5. Progettazione

Tipo di architettura: **Cloud**

Tecnologie client: HTML, CSS, Javascript

Tecnologie server: Node.js, MySQL

Backup: Ogni notte tranne il weekend, tutte le ordinazioni del giorno vengono cancellate la sera in cui non c'è nessuna attività.

5.1 Server

Il server dovrà gestire tutte le richieste:

- **Telegram:** l'utente farà una richiesta unica quando confermerà l'ordine.
- **Sito gestione:** da questo sito potrò aggiungere degli ordini, modificare le classi, le password, rimuovere gli utenti (id telegram).
- **Sito web bar:** il server fa manda la richiesta al sito web o il sito web in polling controlla se c'è un nuovo ordine?

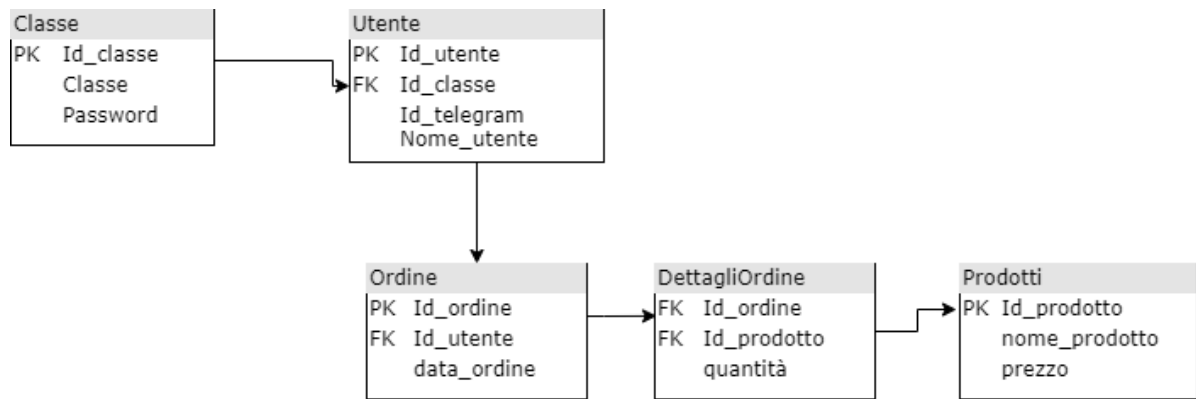
5.2 Database

Database utilizzato: **MySQL** (potrebbe cambiare in base al cloud)

La tabella backup di tutti gli ordini mi servirà per fare un'analisi dei dati per vedere quali sono i prodotti maggiormente consumati.

++DA MODIFICARE PERCHE' MANCA LA TABELLA PER IL BACKUP DEGLI ORDINI++

5.2.1 Schema relazionale



5.2.2 Schema logico

Classe (Id_classe, classe, password)

Utenti (Id_utente, Id_telegram, Id_classe*, nome_utente)

Ordini (Id_ordine, Id_utente*, data_ordine)

Dettagli_ordini (Id_ordine*, Id_prodotto *, quantità)

Prodotti (Id_prodotto, nome_prodotto, prezzo)

5.3 Telegram

Linguaggio utilizzato: **Javascript**.

Pacchetto utilizzato per telegram: **Telegraf**.

Comandi telegram: `"/start"`, `"/logout"`.

5.3.1 Flusso procedurale

- L'utente al primo accesso dovrà inserire la classe e la **password** e poi il suo nome e cognome. Ogni classe avrà una password. Dovrà essere criptata. Il suo account telegram verrà associato ad un id, il login verrà effettuato solo una volta.
- Inizialmente ci saranno due bottoni: **ordina** o **visualizza ordine e modifica**.
- Se preme **ordina** compaiono tutte le ordinazioni sotto forma di **bottone**. L'utente preme il bottone, quindi l'ordine, e lo aggiungerà alla sua lista.
- Se preme **visualizza ordine** potrà visualizzare il suo ordine.
- L'utente può **modificare l'ordine**, potrà eliminare e/o aggiungere alla lista un nuovo ordine. Ogni prodotto sarà numerato e ci sarà un bottone per ogni numero. Per esempio: 1- Panino cotoletta, l'utente preme il bottone 1 elimina quel prodotto dalla lista.
- Una volta che l'utente ha concluso l'ordine, premerà il bottone **invia**. Una volta inviato non sarà più modificabile.

5.3.2 Problemi

- Fino a che l'utente non conferma l'ordine dove vengono salvati i dati?
- Aggiunta di bottoni dinamici su telegram.

5.3.3 Considerazioni

Se salvo i dati localmente su un json dovrò sovrascrivere nel caso in cui volessi fare modifiche.

L'utente farà una **chiamata HTTP** solo quando confermerà l'ordine.

Prendo prezzo e quantità per ogni ordine e poi lato client faccio la somma.

5.4 Sito gestione

Qui l'utente amministratore da un'interfaccia user friendly potrà:

- Aggiungere/rimuovere i prodotti;
- Cambiare il prezzo dei prodotti;
- Aggiungere/rimuovere le classi (questa funzionalità può servire all'inizio dell'anno);
- Aggiungere/rimuovere le persone (nel caso in cui qualcuno si sia impossessato della password della classe);
- Cambiare password per le classi;

5.5 Sito web bar

L'idea è di mettere un computer (raspberry con monitor o un All In One) al bar. Qui girerà il sito web.

Da questo sito web si potrà vedere:

- Le ordinazioni divise per classe
- Le varie statistiche sui prodotti consumati
- Check per dire che l'ordinazione è stata preparata