

Documentazione

Benedetta Vitale - Emilio Meroni

November 19, 2023

Contents

1	Project Plane	5
1.1	Introduzione	5
1.2	Modello di Processo	6
1.3	Organizzazione del Progetto	6
1.4	Standard, Lineeguida, Procedure	6
1.5	Attività di Gestione	7
1.6	Rischi	7
1.7	Personale	7
1.8	Metodi e Tecniche	7
1.9	Garanzia e Qualità	7
1.10	Workpackages	8
1.11	Risorse	8
1.12	Budget	8
1.13	Cambiamenti	8
1.14	Consegna	8

Chapter 1

Project Plane

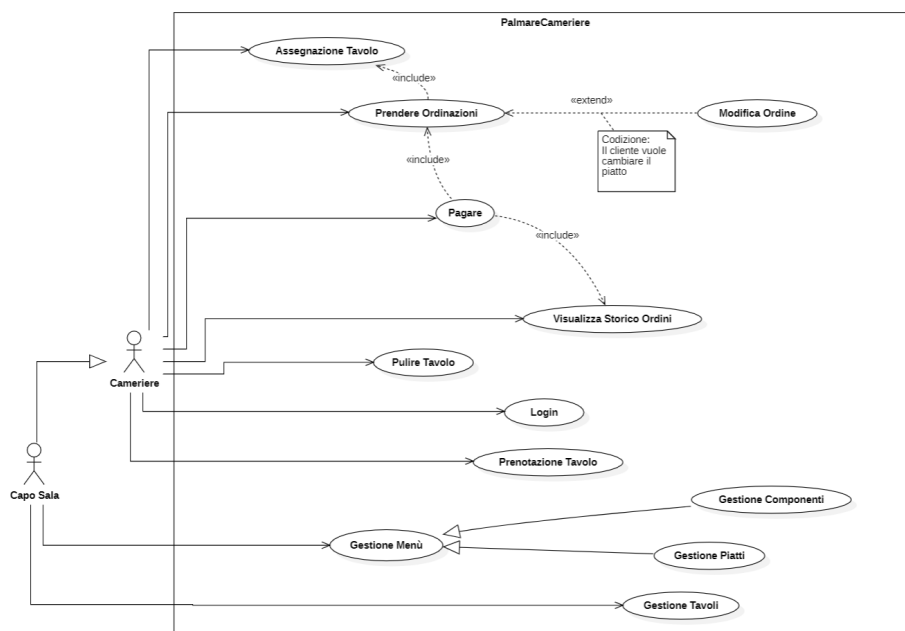


Figure 1.1: Use Case Diagram

1.1 Introduzione

Questo progetto verrà svolto da Benedetta Vitale ed Emilio Meroni, entrambi studenti al terzo anno di ingegneria informatica presso l'università di Bergamo.

Il software che andremo a sviluppare è pensato per dare supporto alla attività di ristorazione. In particolare, dovrà assistere alle mansioni dei camerieri, come

ad esempio prendere le ordinazioni ai tavoli, redigere il conto, trovare tavoli disponibili, ecc.

Abbiamo scelto questa tipologia di sistema dato che Emilio lavora, nei week-end, presso un ristorante, e gli ha incuriosito la gestione interna tramite l'utilizzo dei palmari da parte dei camerieri.

1.2 Modello di Processo

Il modello di processo che abbiamo deciso di seguire è quello della prototipazione, in particolare la prototipazione evolutiva, molto utile per quanto riguarda la costruzione dell'interfaccia grafica, la parte principale della nostra applicazione

1.3 Organizzazione del Progetto

Utilizzeremo una organizzazione a tre livelli:

1. Livello Data Base
2. Livello Logico
3. Livello Presentazione

Il *livello Data Base*, come da nome, si occuperà della parte del DB, il quale sarà embedded, SQLite. Il *livello Presentazione* sarà quello visto dall'utente che usufruirà dell'applicazione, possiamo definirlo come il livello più esterno dove i dati verranno presentati in modo grafico e la gestione dei tavoli e delle comande è fatto in modo interattivo. L'ultimo livello è quello *Logico*, il quale possiamo posizionarlo graficamente tra il livello Data Base e quello Presentazione, funge da intermediario tra i due livelli e fornisce gli oggetti principali per la gestione dell'applicazione

1.4 Standard, Lineeguida, Procedure

Come standard abbiamo scelto di utilizzare lo standard definito da *Java*, il quale si può trovare su questo

1.5 Attività di Gestione

1.6 Rischi

1.7 Personale

1.8 Metodi e Tecniche

1.9 Garanzia e Qualità

Il programma dovrà essere utilizzato in ristoranti, e in particolare dal personale che prenderà le ordinazioni, quindi dovrà essere di *facile* utilizzo, con un focus maggiore sulla funzionalità che sull'estetica. Un'altra qualità che dovrà garantire è la *prevenzione di errori* da parte degli utenti, come ad esempio il "miss-click" (click errati o accidentali). In particolare abbiamo evidenziato quattro qualità che sistema dovrà possedere, di cui, per ciascuna indichiamo come garantiremo che verrà seguito:

1. **Semplice:** programma sarà scritto per avere poche sezioni scritte e di facile comprensione, anche per chi si avvicina al programma per la prima volta. Si utilizzeranno poche schermate che contengono tutto il necessario per le *macro* operazioni definite nelle specifiche dei requisiti e indicate nel case diagram [Figura: 1.1].
2. **Intuibile:** Utilizzo dei colori per indicare gli stati dei *tavoli*, in particolare si è deciso che per i *tavoli liberi* si utilizzerà il colore verde, per i *tavoli occupati* il colore rosso e per i *tavoli da pulire* l'arancione



Figure 1.2: Esempio Bottoni

3. **Prevenzione Errori:** feedback visivo

4. **Veloce:**

È semplice e intuibile perché ha poche scritte di facile comprensione, anche per chi si avvicina al programma per la prima volta al programma; ci sono poche schermate che hanno al loro interno tutto il necessario per le operazioni che si desiderano; inoltre, i bottoni e i colori rendono il programma semplice e intuibile. È veloce in quanto il programma ha poche sezioni in cui bisogna utilizzare la tastiera, mentre il programma è più focalizzato sull'uso dei bottoni che

lo rendono molto più veloce nelle azioni che si dovranno svolgere. Il programma è Prevenzione Errori dato che, con le schermate pop-up di conferma, c'è una riduzione dei possibili errori che si possono creare nell'uso del programma.

1.10 Workpackages

1.11 Risorse

1.12 Budget

1.13 Cambiamenti

1.14 Consegna

In questa sezione si discutono i metodi e le scadenze di consegna del progetto, in particolare la consegna si dividerà in due fasi:

- Consegna del *Project Plane*, il quale dovrà essere consegnato circa un mese prima del primo esame scritto, che si svolgerà nel mese di gennaio; quindi, per il mese di dicembre si dovrà effettuare la prima consegna.
- Consegna del *Progetto*, quest'ultimo avrà una scadenza più lunga, infatti, l'ultimo giorno di consegna sarà cinque giorni prima dell'esame orale.

Per quanto riguarda i metodi di consegna si dovrà condividere con il professore Gargantini la repository di github contenente il progetto, per la consegna di questo documento si dovrà indicare nel file reame, della repository, la posizione del project plane. Mentre per la consegna del progetto si dovrà creare un issue, intitolata “Approvazione Progetto” ed assegnarla al professore.