

**A.A. 2020/2021**

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Processo di sviluppo</b>	<b>3</b>
<b>3</b>	<b>Documento di visione</b>	<b>4</b>
3.1	Introduzione . . . . .	4
3.2	Parti interessate . . . . .	5
3.3	Utenti . . . . .	5
3.4	Obiettivi delle parti interessate . . . . .	5
3.5	Caratteristiche del sistema . . . . .	5
<b>4</b>	<b>Specifica dei Requisiti</b>	<b>7</b>
4.1	Attori Primari . . . . .	7
4.2	Attori Finali . . . . .	8
4.3	Tabella Attori-Obiettivi . . . . .	9
4.4	Diagramma dei Casi D'Uso . . . . .	10
4.4.1	Descrizione Breve . . . . .	10
4.4.2	Descrizione Dettagliata . . . . .	12
4.5	Requisiti non funzionali . . . . .	16
4.5.1	Funzionalità . . . . .	16
4.5.2	Usabilità . . . . .	16
4.5.3	Affidabilità . . . . .	16
4.5.4	Prestazioni . . . . .	16
4.5.5	Sostenibilità . . . . .	16
<b>5</b>	<b>Analisi dei requisiti</b>	<b>18</b>
5.1	Modello di Dominio . . . . .	18
5.2	State Chart Diagram . . . . .	19
5.2.1	Tavolo . . . . .	20

5.2.2	Prodotto Ordinato . . . . .	21
5.2.3	Ordine . . . . .	22
5.3	Sequence Diagram di Dominio . . . . .	22
5.3.1	Gestisci Ordinazioni . . . . .	23
5.3.2	Notifica Ordinazione . . . . .	28
<b>6</b>	<b>Progettazione</b>	<b>29</b>
6.1	Architettura . . . . .	29
6.2	Component Diagram . . . . .	30
6.3	Database . . . . .	31
6.4	Deployment Diagram . . . . .	32
6.5	Class Diagram . . . . .	32
6.6	Sequence Diagram . . . . .	32
<b>7</b>	<b>Implementazione</b>	<b>33</b>

# Capitolo 1

## Introduzione

Il seguente documento descrive il processo di realizzazione di un applicazione gestionale per un locale di ristorazione.

Il sistema realizzato ha come obbiettivo quello di semplificare il lavoro dei dipendenti e la gestione del locale da parte del proprietario . Le principali figure che traggono vantaggio dall'applicazione sono :

- Camerieri : possono prendere le ordinazioni dei clienti in modo del tutto automatizzato sfruttando un dispositivo mobile;
- Proprietario : può controllare la sua attività in tempo reale e ricavare statistiche dai dati memorizzati (i.e. vendite, merce consumata etc.) .
- Addetto all'accoglienza : può gestire lo stato dei tavoli.
- Chef/Pizzaiolo : può visualizzare gli ordini da preparare e segnalare quando ha terminato il suo compito.

Pertanto, i camerieri disporranno di una applicazione da eseguire su smartphone/-tablet mentre i restanti dipendenti del locale avranno un'applicazione desktop con interfacce diverse in base al ruolo che essi ricoprono. In seguito saranno descritti il processo di sviluppo adottato dal team, l'analisi e specifica dei requisiti, le scelte di progetto ed i dettagli implementativi .

## Capitolo 2

### Processo di sviluppo

# Capitolo 3

## Documento di visione

### 3.1 Introduzione

Si vuole realizzare un sistema per la gestione degli ordini e dei pagamenti di un locale di ristorazione.

L'obiettivo è automatizzare e semplificare la gestione delle attività di routine del personale, a partire dalle ordinazioni dei clienti fino ad arrivare all'organizzazione della cucina, della sala e dei pagamenti.

In particolare si vuole far fronte a tutte quelle problematiche legate alle :

- Ordinazioni dei clienti.
- Amministrazione della sala.
- Sincronizzazione tra le varie attività all'interno del locale (cucina, bar, forno, etc.).
- Gestione delle merci a disposizione.
- Pagamenti dei clienti a fine servizio.
- Storici dei dati del locale.

## 3.2 Parti interessate

1. Proprietario del locale.
2. Dipendenti del locale (camerieri, pizzaioli, chef, etc.).
3. Clienti del locale.

## 3.3 Utenti

- Dipendenti di sala (camerieri, addetto all'accoglienza)
- Dipendenti di cucina/forno (chef, pizzaiolo).
- Proprietario del locale.
- Gestore delle merci (economo).

## 3.4 Obiettivi delle parti interessate

1. Un servizio più efficiente e automatizzato massimizza il numero di clienti serviti e la loro soddisfazione, minimizzando errori di distrazione da parte dei dipendenti.
2. Il lavoro del dipendente viene semplificato automatizzando la gestione delle ordinazioni al sistema.
3. I clienti avranno un'esperienza migliore perché verranno ridotti i tempi di attesa e possibili errori dovuti da ordinazioni errate.

## 3.5 Caratteristiche del sistema

Il sistema sarà in esecuzione su un'applicazione "mobile" (su smartphone o tablet) direttamente accessibile dai dipendenti del locale, ma sarà in esecuzione anche su una piattaforma "desktop", o dedicata, per gestire le richieste degli utenti che non hanno la necessità di lasciare la loro postazione durante il servizio.

In primo luogo il dipendente della sala dovrà essere in grado di registrare le ordinazioni dei clienti nel sistema il quale provvederà ad inviarle alle attività del locale interessate (cucina, forno o bar) . In secondo luogo il sistema dovrà essere accessibile

dagli altri dipendenti, come dipendenti della cucina e del bar, per visualizzare le ordinazioni dei clienti registrate recentemente.

Il sistema dovrà permettere di gestire il coordinamento tra reparti diversi attraverso l'uso di messaggi di notifica ed inoltre dovrà fornire da supporto per l'assegnazione dei tavoli. Infine grazie alla memorizzazione di tutte le transazioni il proprietario potrà conoscere i guadagni del locale, la merce consumata e ricavarne statistiche.



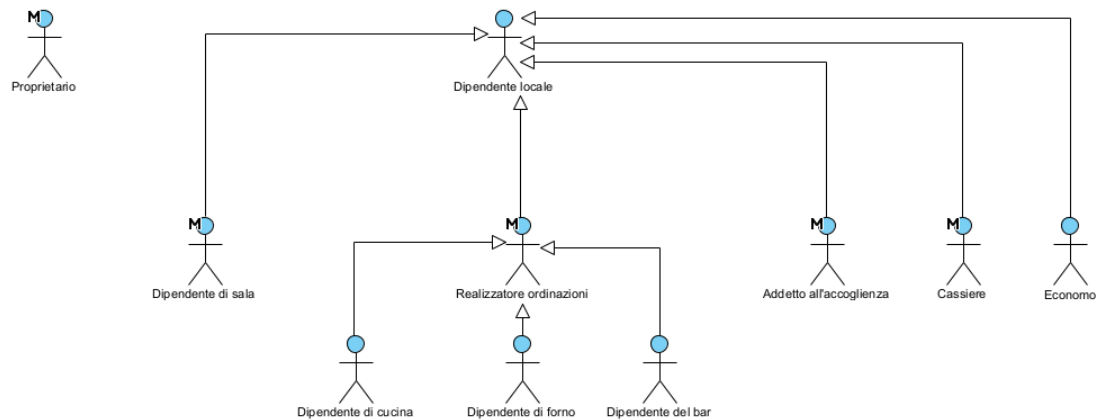
# Capitolo 4

## Specifica dei Requisiti

Per avere una visione chiara di ciò che si deve costruire, quindi una descrizione dettagliata dei requisiti, sono stati effettuati dei colloqui con degli stakeholders e delle sessioni di "brainstorming" tra i componenti del gruppo. In seguito si è costruito un diagramma dei casi d'uso i quali sono stati poi descritti dettagliatamente.

Prima di elencare e descrivere i casi d'uso, è necessario identificare gli attori che caratterizzeranno il sistema.

### 4.1 Attori Primari



- Dipendente di sala: deve interagire con il cliente e prendere le ordinazioni in maniera tale che vengano smistate, tramite il sistema alle varie attività del

locale;

- Dipendenti di cucina/forno/bar: usano il sistema per ricevere i vari ordini e per notificare un completamento di essi;
- Addetto all'accoglienza: usa il sistema per visualizzare i tavoli disponibili ed assegnarli ai clienti;
- Cassiere: richiede al sistema il conto totale di un cliente interessato per completare il pagamento;
- Economo: si occupa della gestione delle merci ovvero l'aggiornamento delle quantità presenti in magazzino (in base alle vendite effettuate dalla sala e dalle richieste dei realizzatori di ordinazioni);
- Proprietario: può usare il sistema per aggiungere e rimuovere dipendenti, consultare dati come vendite, guadagni, quantità di merci.

Nella realtà non tutti gli attori descritti corrisponderanno a persone fisiche differenti, infatti alcuni dipendenti potranno assumere le funzionalità di più attori (ad esempio un dipendente di sala potrebbe ricoprire anche il ruolo di addetto all'accoglienza). A tal proposito quindi si è scelto di intendere gli attori come **ruoli** che un dipendente reale possa ricoprire.

## 4.2 Attori Finali



- Cliente: pur non interagendo con il sistema, è direttamente interessato in alcuni suoi casi d'uso in quanto gli consentono di usufruire del servizio del locale

### 4.3 Tabella Attori-Obiettivi

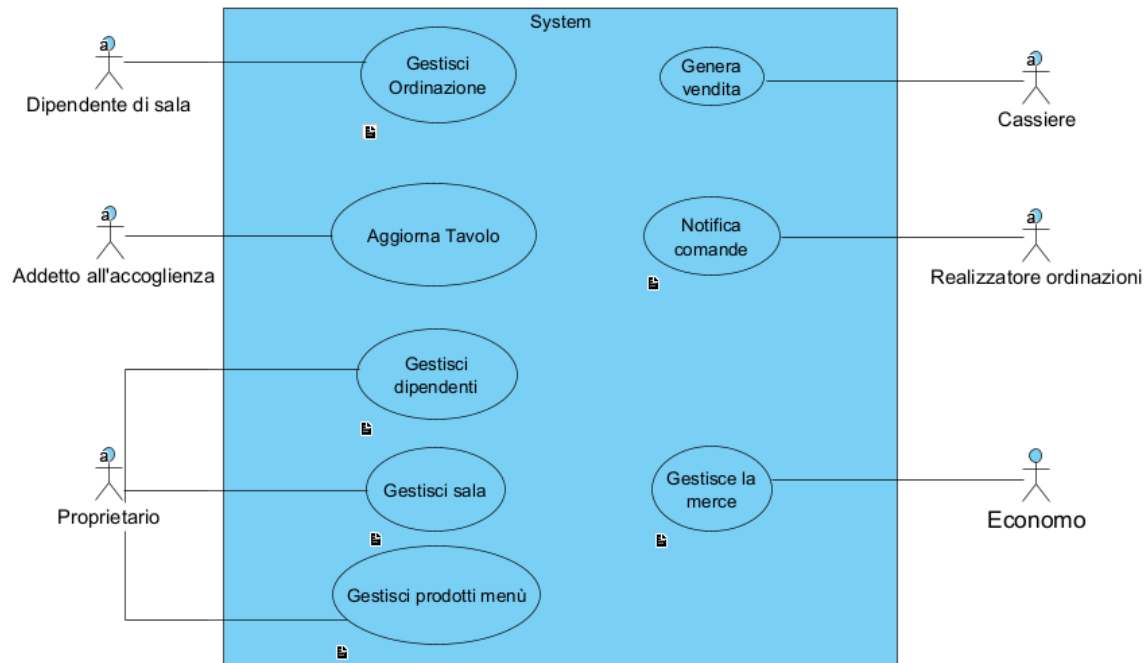
Dopo aver identificato a pieno gli attori che utilizzeranno il sistema, bisogna associare ad ogni attore i suoi obiettivi.

Attore	Obiettivi
Proprietario	Inserimento di un dipendente Rimozione di un dipendente Modifica del ruolo di un dipendente Visualizzazione dei dipendenti Inserimento di un tavolo Rimozione di un tavolo Modifica di un tavolo Visualizzazione dei tavoli Inserimento di prodotti nel menu Rimozioni di prodotti dal menu Modifica di prodotti del menu Visualizzazione del menu completo
Dipendente di sala	Creazione di un'ordinazione per un cliente Rimozione di un'ordinazione Modifica di un'ordinazione Visualizzazione delle ordinazioni
Accoglienza	Assegnazione di un tavolo a dei clienti Modifica dello stato di un tavolo (libero, occupato, riservato, ecc.) Visualizzazione di tutti i tavoli
Cassiere	Generazione vendita di un tavolo (pagamento dei clienti)
Realizzatore di ordinazione	Visualizzazione degli ordini per area (cucina, bar, forno) Notifica completamento ordinazione
Economo	Inserimento di una merce Modifica di una merce Rimozione di una merce Visualizzazione di una merce

Gran parte degli obiettivi riguardano operazioni *CRUD* (Create, Read, Update, Delete) che possono essere raggruppate in un solo obiettivo.

## 4.4 Diagramma dei Casi D'Uso

Tutti gli obiettivi CRUD dei vari attori sono stati inglobati in un unico caso d'uso con il prefisso **Gestisci**, per indicare le operazioni che racchiudono.



### 4.4.1 Descrizione Breve

#### Gestisci Ordinazione

Come descritto in precedenza questo caso d'uso racchiude le operazioni CRUD riguardo ad un'ordinazione di uno o più clienti. Il cameriere infatti deve essere in grado di creare un'ordinazione per un cliente e poterla in seguito modificare, visualizzare o addirittura rimuovere dal sistema.

#### Aggiorna Tavolo

L'addetto all'accoglienza si occupa di far accomodare i clienti nella sala. Il suo obiettivo è quindi avere una visione completa dei tavoli comunicando tramite il sistema

quali tavoli hanno clienti in attesa di ordinazione, quali sono liberi e quali sono occupati.

### **Gestisci Dipendenti**

Il proprietario può aggiungere/rimuovere dipendenti dal sistema ed inoltre è in grado di decidere/modificare i ruoli o il ruolo di ogni dipendente durante il servizio.

### **Gestisci Sala**

Il proprietario può indicare l'identificativo di ogni tavolo in relazione alla sala in cui appartiene. Gli altri dipendenti quindi posso accedere tramite l'identificativo a tutte le informazioni che riguardano i tavoli (numero di persone accomodate, lista dei prodotti ordinati, ecc).

### **Gestisci Prodotti Menù**

Il proprietario può effettuare operazioni CRUD per la gestione del menù del locale.

### **Genera Vendita**

Il compito del cassiere è quello di registrare il pagamento dei clienti in relazione a ciò che hanno ordinato. Esso quindi deve poter risalire tramite il tavolo a tutti gli ordini associati ai clienti per ricavare il prezzo complessivo da pagare.

### **Notifica Ordinazione**

Per rendere il sistema quanto più automatizzato possibile, i realizzatori di ordinazioni (cuoco, pizzaiolo, etc.) devono notificare il completamento di un'ordinazione tramite il sistema. In questo modo, inviando una notifica il sistema provvederà ad avvisare altri dipendenti in attesa di quel completamento.

### **Gestisci Merce**

Ogni prodotto del menu è composto da un insieme di merci disponibili in magazzino. Il compito dell'economo è quindi quello di gestire le merci, per avere sotto controllo le merci in esaurimento. L'esaurimento di una merce comporta la non disponibilità di tutti i prodotti del menu che la contenevano.

#### 4.4.2 Descrizione Dettagliata

La descrizione dettagliata è stata fatta solo per i casi d'uso di rilievo.

<b>Titolo</b>	Gestisci Ordinazione
<b>Livello</b>	Obiettivo Utente
<b>Portata</b>	Applicazione Android o interfaccia Desktop
<b>Attore Primario</b>	Dipendente di sala
<b>Parti Interessate</b>	<ul style="list-style-type: none"> <li>–<u>Dipendenti di sala</u> (vogliono eseguire le loro mansioni)</li> <li>–<u>Proprietario</u> (vuole che i clienti siano soddisfatti dal servizio)</li> <li>–<u>Clienti</u> (vogliono essere serviti nel miglior modo possibile)</li> <li>–<u>Realizzatore di ordinazioni</u> (vuole ricevere le ordinazioni)</li> </ul>
<b>Pre-Condizioni</b>	<ul style="list-style-type: none"> <li>–Il dipendente di sala non può gestire ordinazioni dei tavoli non suoi</li> <li>–Il dipendente di sala non può aggiungere prodotti ad ordinazioni già completate</li> <li>–Il dipendente di sala non può rimuovere prodotti in lavorazione di una ordinazione</li> <li>–Il dipendente di sala non può rimuovere merce ad un prodotto in lavorazione</li> </ul>
<b>Garanzia di successo</b>	<ul style="list-style-type: none"> <li>–Una ordinazione viene creata/modificata/rimossa</li> <li>–Una notifica viene inviata ad un realizzatore di ordinazione</li> <li>–La merce numerabile costituente i prodotti viene riservata per l'ordinazione</li> </ul>
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. Dipendente di sala visualizza i tavoli in attesa di ordinazione di una sala</li> <li>2. Dipendente di sala seleziona un tavolo</li> <li>3. Dipendente di sala visualizza ordinazioni tavolo</li> <li>4. Dipendente di sala visualizza ordinazione</li> </ol>

**Estensioni**

- 3.A Dipendente di sala inizia ordinazione a tavolo
- 4.A Cliente comunica i prodotti
- 5.A Il dipendente di sala inserisce prodotti con eventuali merci aggiuntive
- 6.A Il cliente comunica preferenza di consegna per prodotti
- 7.A Il dipendente di sala inserisce la preferenza di consegna
- 8.A Il dipendente di sala aggiunge ordinazione
- 9.A Il sistema viene aggiornato
- 10.A Vengono inviate notifiche in maniera intelligente ai realizzatori di ordinazioni
  
- 3.B Il dipendente di sala rimuove una ordinazione
- 4.B Il sistema viene aggiornato
- 5.B Viene notificato l'evento ad eventuali realizzatori precedentemente notificati
  
- 4.C Cliente comunica al dipendente di modificare un prodotto all'interno di un'ordinazione
- 5.C Dipendente di sala modifica l'ordinazione con i nuovi prodotti
- 6.C Dipendente di sala inoltra la modifica
- 7.C Il sistema inoltra la modifica ai corrispondenti realizzatori
  
- 4.D Cliente comunica al dipendente di aggiungere un prodotto all'interno di un'ordinazione
- 5.D Dipendente aggiunge il prodotto (o i prodotti) all'ordinazione
- 6.D Il sistema viene aggiornato e invia una notifica ai realizzatori
  
- 4.E Cliente comunica al dipendente di rimuovere un prodotto all'interno di un'ordinazione
- 5.E Dipendente rimuove il prodotto (o i prodotti) all'ordinazione

	6.E Il sistema viene aggiornato e invia una notifica ai realizzatori
<b>Scenari Alternativi</b>	4.A.1 Prodotto comunicato non è disponibile 4.A.2 Ripeti 4.A  3.B.1 Ordinazione non rimovibile (già completata) 3.B.2 Ripeti 3.B 4.C.1 L'ordinazione non è modificabile (poichè in lavorazione) 4.C.2 Ripeti 4.C  4.D.1 Il prodotto non è disponibile 4.D.2 Ripeti 4.D  4.E.1 Il prodotto non è rimovibile (poichè in lavorazione) 4.E.2 Ripeti 4.E

<b>Titolo</b>	Notifica Ordinazione
<b>Livello</b>	Obiettivo Utente
<b>Portata</b>	Applicazione Android o interfaccia Desktop
<b>Attore Primario</b>	Realizzatore di ordinazione
<b>Parti Interessate</b>	– <u>Realizzatore di ordinazione</u> (vuole notificare l'aver completato uno o più lavori, altri realizzatori inoltre possono aspettare questa notifica come segnale per poter iniziare il loro lavoro). – <u>Cliente</u> (vuole ricevere l'ordinazione). – <u>Proprietario</u> (vuole che cliente riceva l'ordinazione)
<b>Pre-Condizioni</b>	–L'ordinazione da notificare esiste



<b>Garanzia di successo</b>	<p>–Prodotto in ordinazione viene posto in stato di terminato</p> <p>–Qualora i prodotti di tutta l'ordinazione vengano posti come terminati allora anche l'ordinazione stessa viene posta come terminata</p>
<b>Scenario Principale</b>	<ol style="list-style-type: none"> <li>1. Realizzatore richiede ordinazioni al sistema</li> <li>2. Il sistema restituisce le ordinazioni che può eseguire</li> <li>3 Il realizzatore seleziona un prodotto</li> <li>4 Il sistema restituisce i dati del prodotto</li> <li>5. Il realizzatore notifica il completamento del prodotto</li> <li>6. Il sistema si aggiorna</li> <li>7. Il sistema sblocca prodotti pendenti</li> </ol>
<b>Estensioni</b>	<ol style="list-style-type: none"> <li>5.A Il prodotto completato era l'ultimo</li> <li>6.A Il sistema aggiorna lo stato dell'ordinazione</li> <li>7.A Il sistema invia un'ordinazione pendente per altri realizzatori di ordinazione in attesa di quel completamento</li> </ol>
<b>Scenari Alternativi</b>	<ol style="list-style-type: none"> <li>7.B.1 Non ci sono ordini pendenti</li> </ol>

## 4.5 Requisiti non funzionali

Il sistema oltre a realizzare i requisiti descritti precedentemente deve rispettare dei vincoli di funzionamento. Tutti i requisiti non funzionali vengono descritti tramite delle specifiche supplementari di tipo **FURPS+**

### 4.5.1 Funzionalità

Per tutti i casi d'uso deve esistere una gestione degli errori ottimale. Durante il funzionamento dell'applicazione, il sistema non deve interrompersi a causa degli errori non gestiti.

### 4.5.2 Usabilità

L'interfaccia utente deve essere semplice e intuitiva permettendo così al dipendente di interagire con il sistema nel modo più veloce possibile.

Per quanto riguarda i camerieri, l'applicativo può essere installato sul proprio dispositivo mobile (ad esempio il proprio smartphone) in modo tale da ridurre al minimo i costi per dispositivi dedicati. Per i dipendenti i quali dispongono di una postazione fissa l'applicativo sarà invece in esecuzione su un computer fisso con monitor touch screen in modo da rendere l'interazione con l'applicazione semplice e veloce.

### 4.5.3 Affidabilità

Il sistema garantirà un corretto coordinamento tra i dipendenti della sala ad esempio impedendo che una stessa ordinazione venga eseguita due volte, tenendo traccia delle ordinazioni gestite da ogni cameriere, .

### 4.5.4 Prestazioni

Il sistema deve funzionare in Real-Time. La gestione delle ordinazioni, soprattutto, deve poter rispondere immediatamente alle variazioni, modifiche e aggiunta di eventuali ordinazioni nel sistema, per notificare velocemente coloro che dovranno realizzarle.

### 4.5.5 Sostenibilità

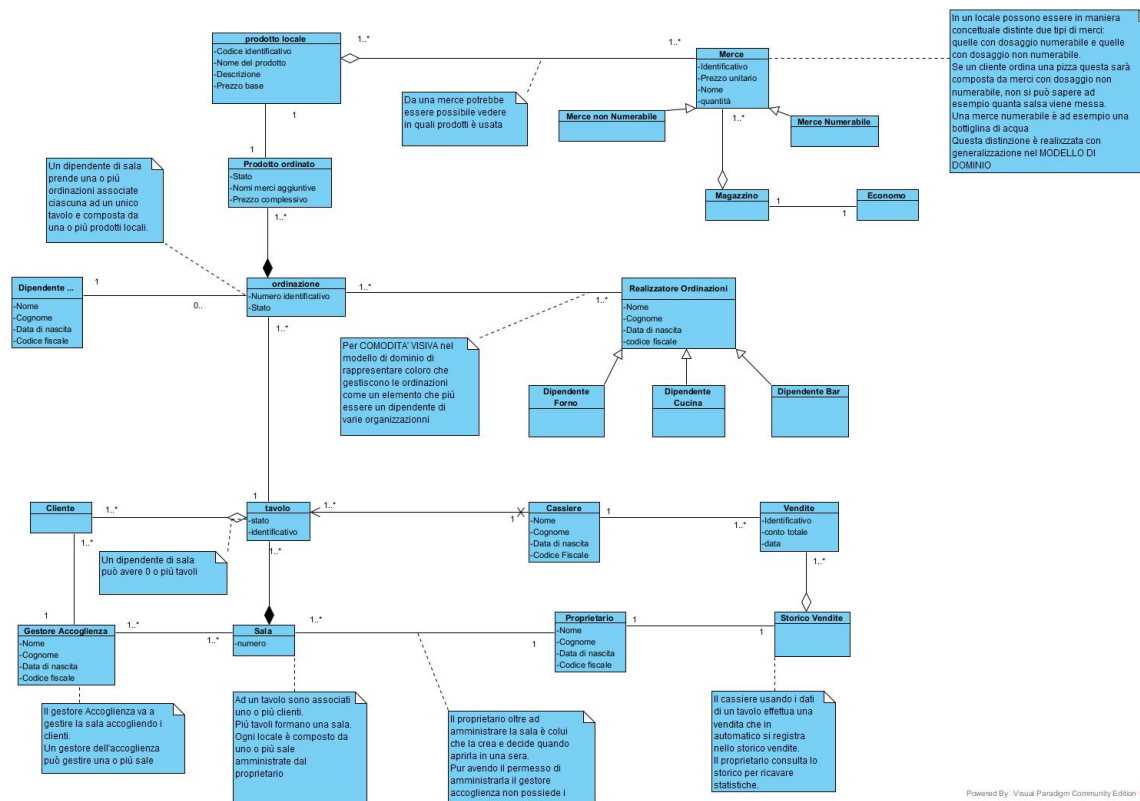
Nel lato "mobile" il sistema deve adattarsi alla maggior parte degli smartphone in commercio, onde evitare problemi di compatibilità.

Nel lato "desktop" il sistema deve essere portatile per poter essere configurato in un qualunque ambiente.

# Capitolo 5

## Analisi dei requisiti

### 5.1 Modello di Dominio



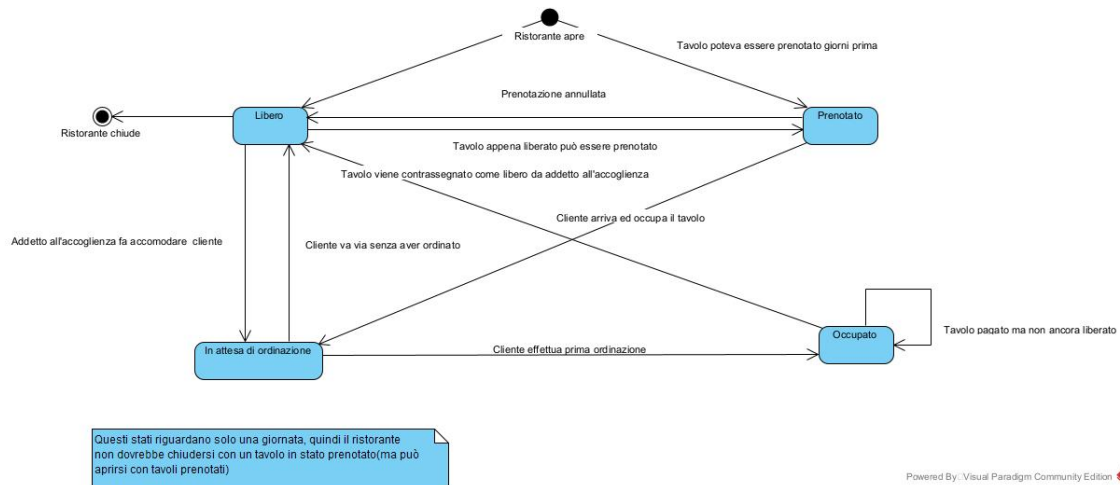
Con questo primo diagramma di analisi si sono fatte le prime ipotesi di funzionamento (descritte dai commenti), mettendo in evidenza le relazioni tra le varie entità del sistema.

1. In un locale con più camerieri si può verificare la situazione in cui diversi camerieri possono prendere le ordinazioni ad uno stesso tavolo. Con questa ipotesi quindi il cameriere non viene associato direttamente al tavolo, ma all'ordinazione che ha creato per esso.
2. Il sistema gestisce le merci in modo automatizzato. Esse sono divise in *numerabili* e *non numerabili*. A tal proposito, quando il cameriere crea un'ordinazione che contiene delle merci numerabili, esse vengono immediatamente scalate dal magazzino.
3. Tutte le vendite completate vengono memorizzate in uno storico.
4. Il prodotto che viene aggiunto all'ordinazione non equivale al prodotto presente nel menu. Esso infatti può subire delle modifiche come aggiunta o rimozioni di merce dalla sua versione di base. Viene indicato quindi come l'entità *Prodotto Ordinato*.

## 5.2 State Chart Diagram

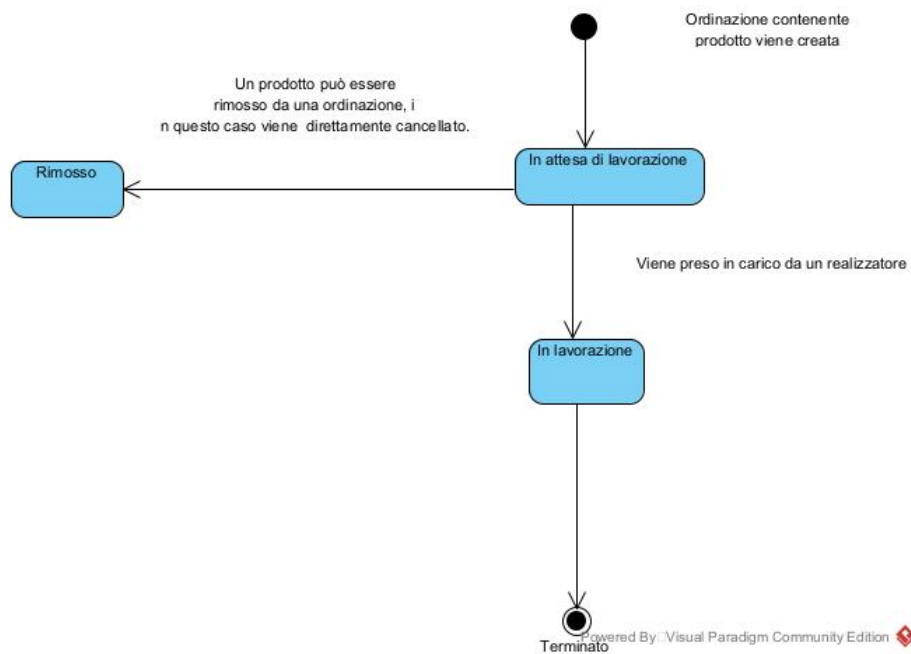
Alcune entità del sistema hanno delle funzionalità diverse in relazione allo stato in cui si trovano.

### 5.2.1 Tavolo



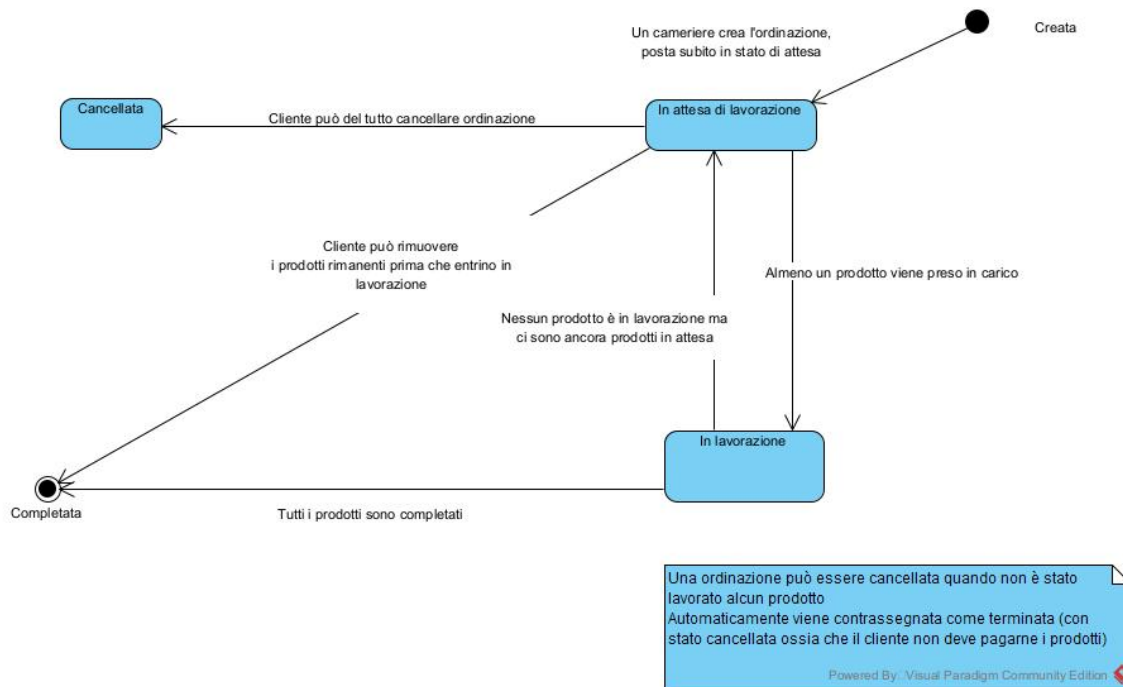
Questi stati valgono solo durante l'esecuzione del sistema. In teoria il locale non può chiudere con tavoli occupati o in attesa di ordinazione. Attraverso questi stati esiste anche più controllo per evitare errori umani. Ad esempio un cameriere non può prendere le ordinazioni di un tavolo libero, ovvero non occupato da nessun cliente.

### 5.2.2 Prodotto Ordinato



Gli stati vengono aggiornati tramite dei feedback di chi realizza i relativi prodotti.

### 5.2.3 Ordine



Gli stati dell'ordine dipendono molto dagli stati dei prodotti ordinati che lo compongono, essendo infatti un'aggregazione di prodotti.

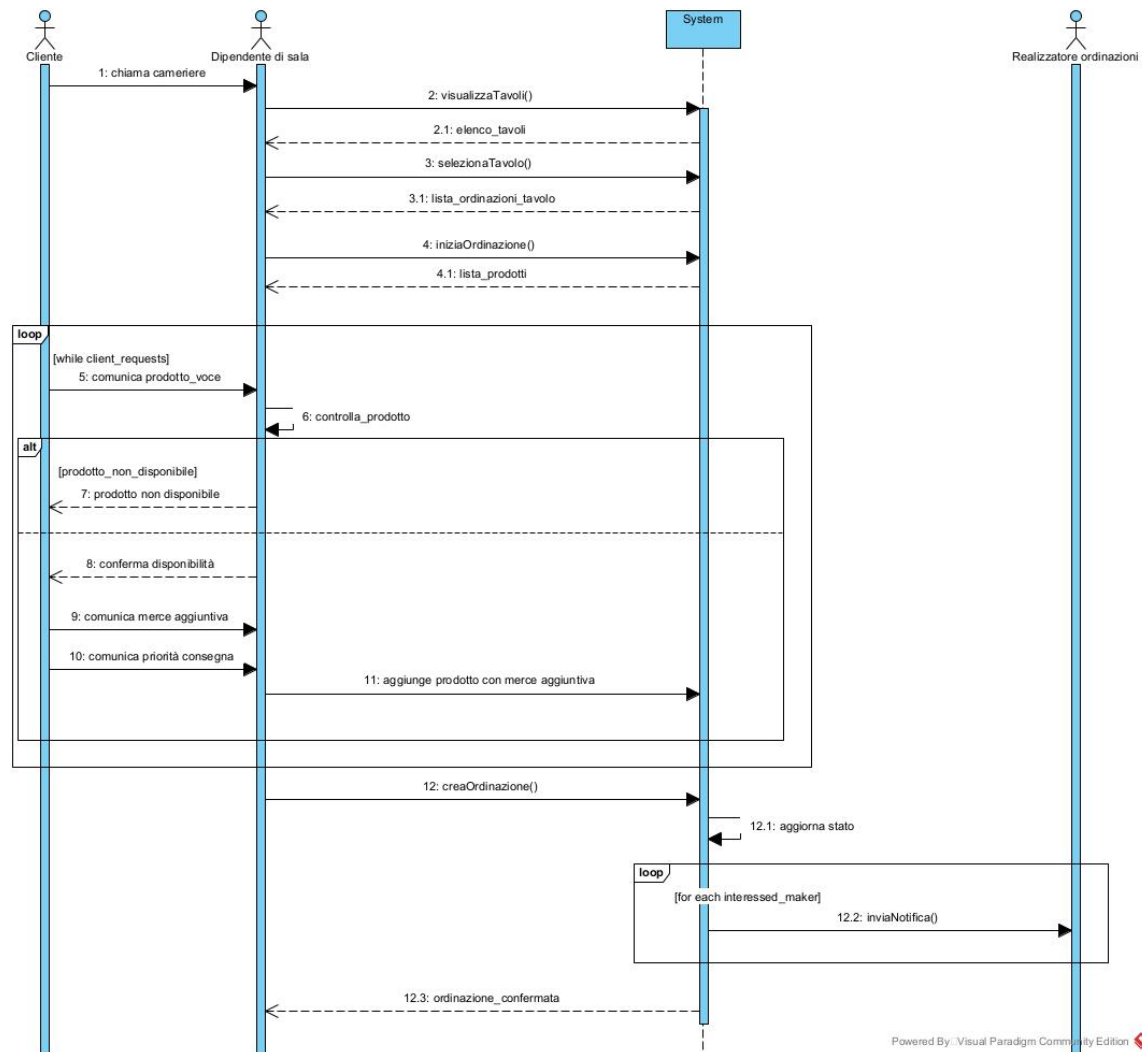
## 5.3 Sequence Diagram di Dominio

I seguenti sequence diagram sono realizzati solo per indicare le interazioni tra un utente e il sistema.



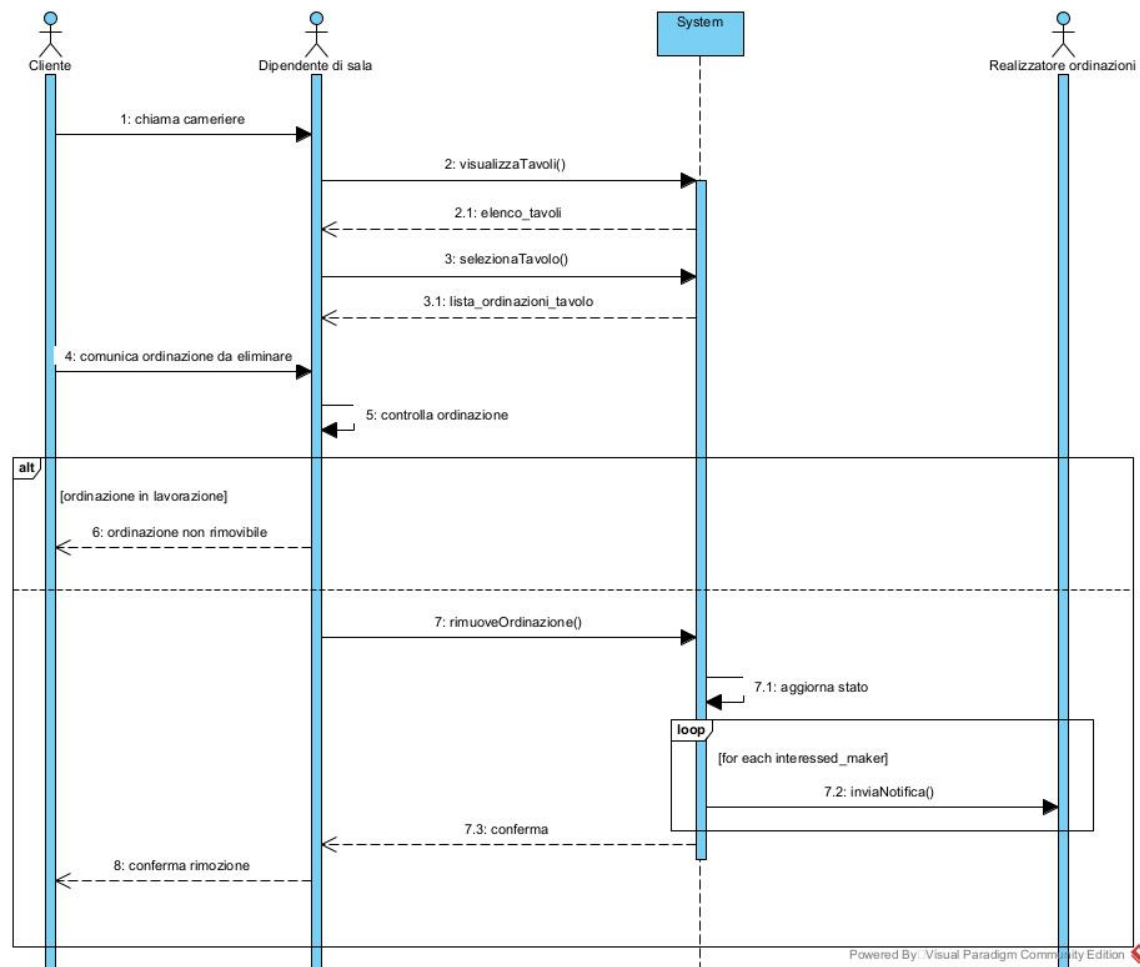
### 5.3.1 Gestisci Ordinazioni

#### Estensione A

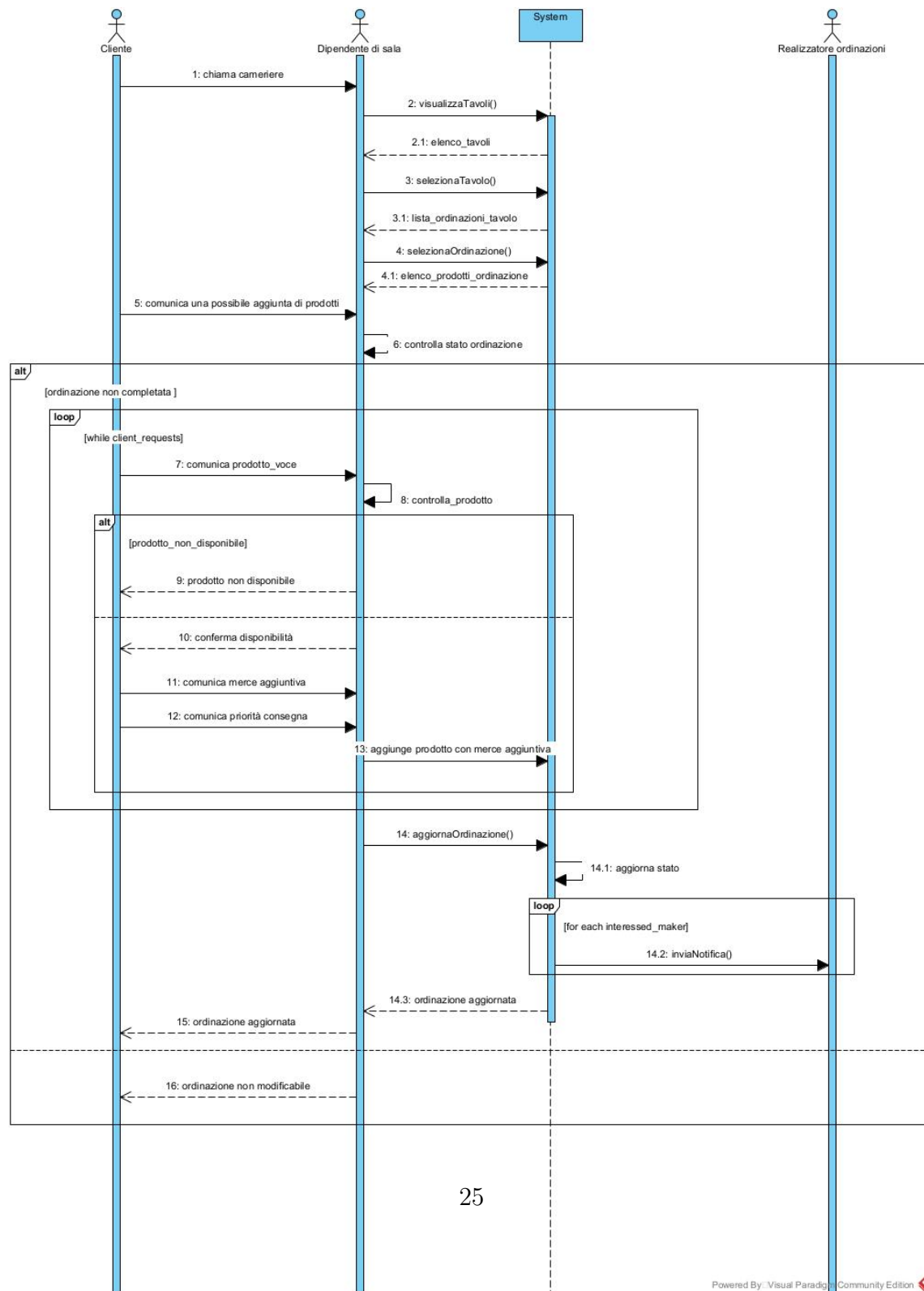


Powered By: Visual Paradigm Community Edition

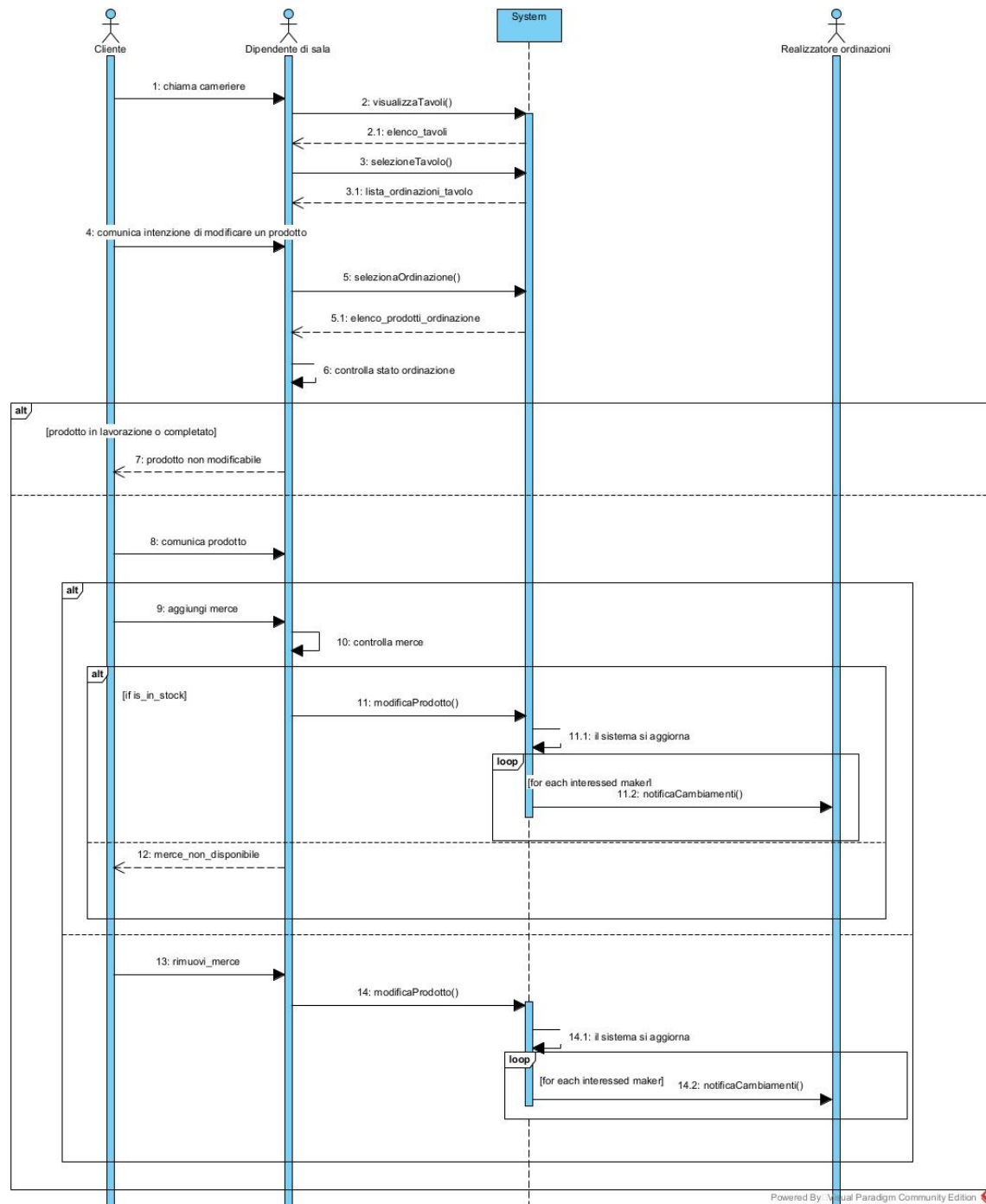
## Estensione B



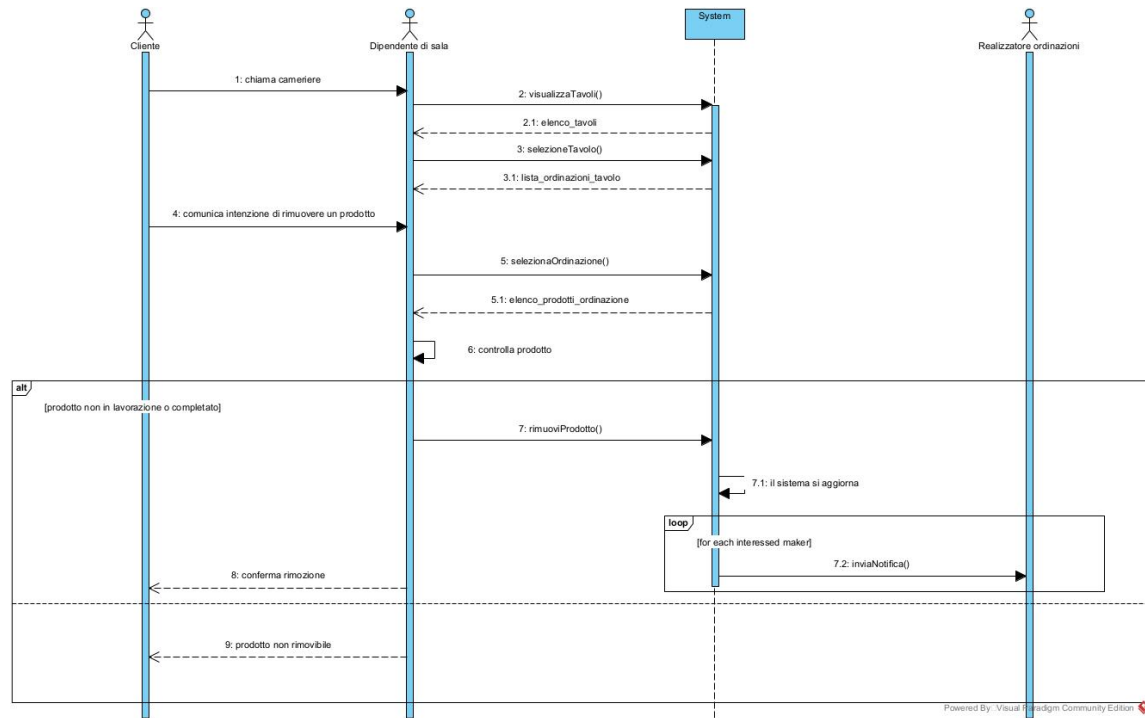
## Estensione C



## Estensione D

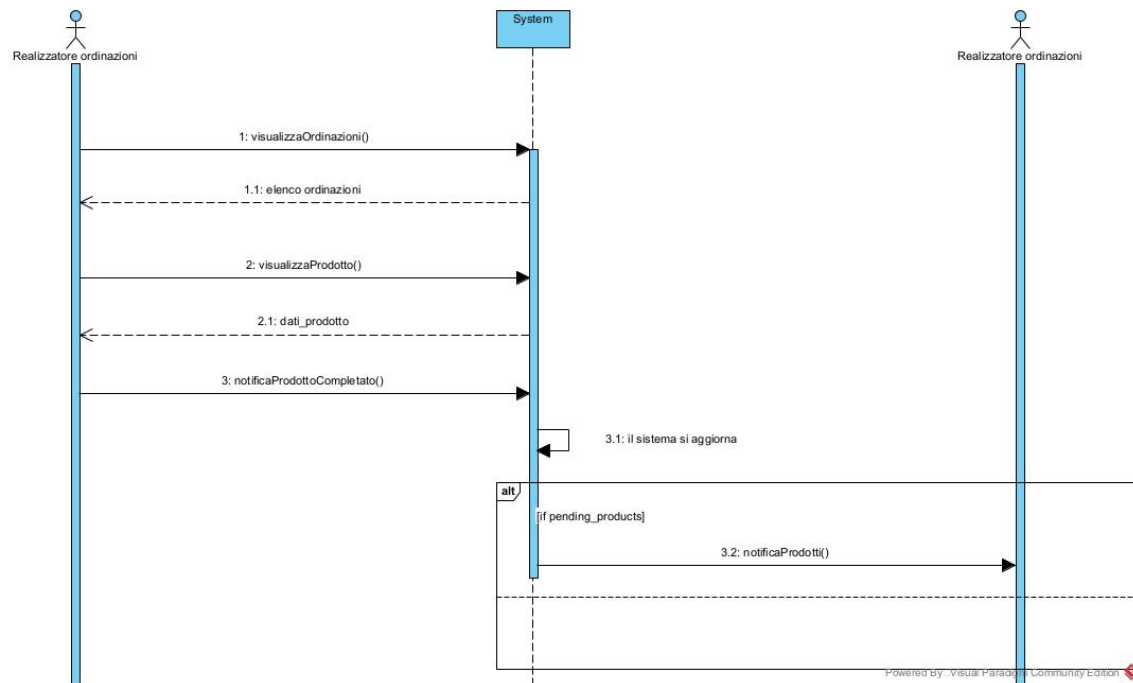


## Estensione E



### 5.3.2 Notifica Ordinazione

#### Estensione A



# Capitolo 6

## Progettazione

La fase di progettazione prevede principalmente la scelta dell'architettura da usare, su cui costruire il sistema.

Per identificare l'architettura ideale bisogna elencare quali sono i componenti che ne fanno parte e in che modo sono interconnessi.

- I componenti sono di due tipologie:
  - Dispositivi utente, rappresentano i dispositivi fisici concessi ai dipendenti (smartphone o applicazione desktop)
  - Sistema centrale, racchiude la logica di gestione e a cui i dispositivi utente possono fare accesso.
- I connettori sono principalmente protocolli di rete, in quanto il sistema è pensato per dipendenti che hanno la possibilità di spostarsi all'interno del locale e accedere al sistema centra in modalità remota.

Alla base di ciò i componenti devono ricevere dei cambiamenti del sistema tramite opportune *notifiche*, evitando quindi attese attive.

### 6.1 Architettura

Lo stile architetturale utilizzato per realizzare il sistema è il **Publish-Subscribe**. Per la realizzazione dell'applicativo infatti vengono ripresi tutti i vantaggi dello stile a *invocazione implicita*:

- Tutti i componenti sono altamente disaccoppiati, favorendo la scalabilità del sistema. Il proprietario può quindi assumere/licenziare i dipendenti, o cambiarne i ruoli senza vincolarsi dal sistema.

- Tutta la comunicazione avviene tramite chiamate invocate implicitamente in risposta ad un evento.

In relazione ai componenti descritti, tutti i dipendenti sono dei *Subscribers* mentre il sistema centrale è l'unico *Publisher*.

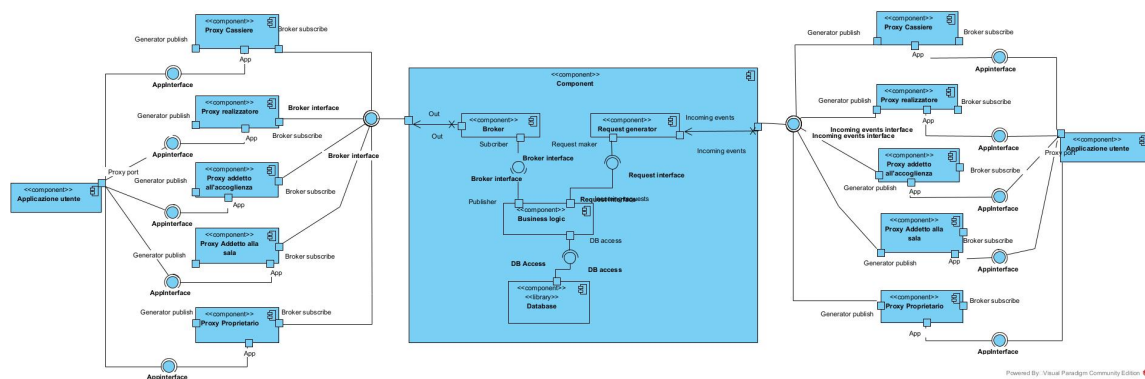
I dipendenti, effettuando l'accesso, si iscrivono in automatico al Publisher. Dato che il sistema può contenere un numero notevole di dipendenti, il Publisher fa uso di **Proxy** per smistare i messaggi solamente a specifici Subscribers.

Essi infatti all'atto del login non conoscono i Proxy del sistema, ma è il sistema stesso che li racchiude in gruppi (con lo stesso ruolo) e li associa a determinati Proxy.

Riassumendo quindi:

<b>Componenti</b>	Publisher, Subscribers, Proxy
<b>Connettori</b>	Protocolli di rete
<b>Dati</b>	Notifiche
<b>Topologia</b>	Subscribers sono connessi al Publisher indirettamente, ricevendo e inviando notifiche attraverso i Proxy
<b>Vantaggi</b>	Subscribers sono tutti disaccoppiati lavorando in parallelo. Le notifiche vengono ben distribuite grazie ai Proxy
<b>Svantaggi</b>	Il controllo computazionale è caricato tutto sul sistema centrale. Nessuna conoscenza di quali componenti risponderanno ad un eventi

### 6.1.1 Component Diagram





Il database deve contenere le informazioni di base che devono essere utilizzate, a partire dai dipendenti che vengono registrati nel sistema e finire con le ordinazioni che vengono effettuate durante l'utilizzo del sistema.



1. Le merci hanno, oltre alle informazioni di base quali prezzo, codice identificativo, nome, etc, hanno un attributo *priceAsAdditive*. Tale attributo, se non nullo, indica il prezzo da aggiungere se quella merce è usata come merce aggiuntiva di un prodotto.
2. Il *Prodotto Ordinato* è un'associazione ternaria tra *Ordine*, *Prodotto* e *Merce*.
3. Ogni prodotto ha un attributo collegato all'entità *Ruoli*. Esso serve per indicare a che attività appartiene quel prodotto.

### **6.3 Deployment Diagram**

### **6.4 Class Diagram**

### **6.5 Sequence Diagram**

# Capitolo 7

## Implementazione

# Bibliografia