



Università di Salerno  
Corso di Ingegneria del Software

ClickEat  
Problem Statement  
Versione 1.5



ClickEat

## Data

Progetto: ClickEat	Versione: 1.4
Documento: Problem Statement	Data: 20/10/2018

## Partecipanti:

Nome	Matricola
Cupito Andrea [CA]	0512104538
Amoriello Luca [AL]	0512104658
Pasquariello Giovanni [PG]	0512105020
Russo Vincenzo [RV]	0512104130

Scritto da:	Cupito Andrea
-------------	---------------

## Revision History

Data	Versione	Descrizione	Autore
12/10/2018	1.0	Descrizione del software da farsi, individuazione obiettivo finale, creazione logo.	[AC], [PG]
13/10/2018	1.1	Requisiti funzionali e suddivisione di essi in macro-categorie	[AL]
14/10/2018	1.2	Requisiti non funzionali	[PG]
15/10/2018	1.3	Aggiunta scenari	[RV]
20/10/2018	1.4	Revisione totale del documento, suddivisione dei requisiti funzionali in macrocategorie	[AL], [AC]

## Sommario

### 1. Introduzione

- 1.1 Scopo del sistema
- 1.2 Ambito del sistema
- 1.3 Obiettivi e criteri del successo del progetto
- 1.4 Definizioni, acronimi e abbreviazioni
- 1.5 Referenze
- 1.6 Overview

### 2. Sistema Corrente

### 3. Sistema Proposto

- 3.1 Panoramica
- 3.2 Requisiti Funzionali
  - RF 1.0 Gestione della sala
  - RF 2.0 Gestione ordinazione
  - RF 3.0 Gestione ristorante
  - RF 4.0 Gestione utenti
- 3.3 Requisiti non funzionali
  - 3.3.a Usabilità
  - 3.3.b Affidabilità
  - 3.3.c Performance
  - 3.3.d Manutenibilità
  - 3.3.e Implementazione
  - 3.3.f Sicurezza

# 1. Introduzione

La piattaforma ClickEat andrà a sostituire l'ormai obsoleto metodo di ordinazione cartaceo attualmente presente nei ristoranti. Il vecchio sistema prevedeva di dover trascrivere l'ordinazione su un blocco cartaceo. Tale sistema espone il personale del ristorante alla perdita della prenotazione siccome è facile che un foglio si perdi o venga buttato; inoltre, è difficile avere una visione completa della sala ad occhio nudo, si intende dire che riconoscere quale tavolo sia libero tra le centinaia di persone sedute, risulta essere complicato.

Inoltre, per ogni ordinazione sono previste diverse portate e queste potrebbero essere prodotte in luoghi differenti (ad esempio un primo piatto verrà preparato in cucina, mentre una pizza verrà preparata in pizzeria). Questo per dire che, per un tavolo, è necessario portare un foglio contenente l'ordinazione ad ogni reparto per far sì che essi vengano svolti tutti. Quindi, da come si evince, il sistema discusso in precedenza richiede tempi abbastanza lunghi ed uno spreco di carta elevato. Inoltre, in ogni reparto, c'è un foglio per ogni ordinazione; ciò comporta spesso confusione, poiché per un movimento brusco i fogli potrebbero spargersi ovunque ed è difficile dopo ristabilire l'ordine con cui sono arrivati i biglietti.

Analizzando queste esigenze, quindi, si è deciso di studiare una soluzione per ottimizzare i tempi, gli sprechi e gli spazi: ClickEat.

Tale sistema permetterà ad un particolare utente (addetto alla sala) di registrare le ordinazioni di un tavolo; allo stesso tempo su un terminale, attraverso l'utilizzo di un'altra porzione del sistema, sarà possibile visualizzare lo stato di tutti i tavoli presenti in sala (occupato/libero) e la loro ordinazione nel caso siano occupati

## 1.1 Scopo del sistema

L'obiettivo del sistema è quello di informatizzare il sistema di ordinazione di un ristorante, sostituendo il metodo cartaceo, con un sistema in grado di aumentare l'efficienza in termini di tempo e di risorse di un'ordinazione prevedendo un tablet al post del classico blocco note e in ogni reparto (cassa compresa) un terminale in sostituzione al "mucchio di carta" per ogni ordinazione. Per efficienza si intende oltre a quella del sistema in sé, anche la riduzione del tempo di consegna della comanda in ogni reparto e la diminuzione del tempo di calcolo della somma del conto.

## 1.2 Ambito del sistema

L'utilizzo del sistema è rivolto a tutte le attività commerciali di ristoro, che effettuano un servizio di ordinazione. Astruendo il concetto di ordinazione, sarebbe possibile utilizzare la piattaforma anche per le ordinazioni in un bar che offre un servizio a tavolino, con le opportune configurazioni.

## 1.3 Obiettivi e criteri del successo del progetto

Per considerare il progetto come utilizzabile dovrebbe rispettare tutte le richieste descritte finora ed eseguire le operazioni in tempi brevi. Un'ordinazione, la stampa di un conto, la modifica di un'ordinazione non deve richiedere più del tempo dovuto.

Ciò non deve escludere la possibilità di interlocuzione tra i vari addetti in un ristorante per la coordinazione di un'ordinazione. Ad esempio, in un caso reale, se un ordine richiede la preparazione di diverse pietanze da reparti diversi, ma che debbano uscire nello stesso momento, il cameriere può coordinare i reparti per far sì che ciò avvenga.

## 1.4 1.4 Definizioni, acronimi e abbreviazioni

Termine	Significato
ClickEat	Nome del sito web e del sistema in sviluppo.
Login	Attività di accesso alla piattaforma tramite le proprie credenziali.
Log out	Attività di uscita dell'account connesso.

## 1.5 Referenze

- Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering - Using UML, Patterns, and Java™, Third Edition, 2009

## 2.Sistema Corrente

Senza un sistema informatizzato per la gestione delle ordinazioni le operazioni che si compiono sono le seguenti:

- Dopo aver fatto accomodare i clienti e aver portato il menù si procede l'ordinazione su un blocco note a due o tre copie. Viene indicato il numero del tavolo e il numero di persone;
- Sempre sul blocco note vengono trascritti, divisi per categoria i prodotti richiesti dai clienti. (Vedere figura 1).
- Dopo aver trascritto tutto, le tre copie vengono ripartite nelle varie sezioni del ristorante:
  - 1 copia viene consegnata alla cucina per preparare antipasti, frittura, primi o secondi;
  - Un'altra copia viene consegnata alla cassa, per poter tenere traccia dell'ordine preso dai tavoli (per effettuare il conto);
  - Un'ulteriore copia, eventualmente, potrà essere consegnata alla pizzeria.

Qualsiasi modifica successiva dell'ordine verrà riportata sulla copia presente alla cassa e nella sezione relativa alla modifica. Man mano che gli ordini verranno evasi nei vari reparti, la copia del tavolo verrà buttato.

N.B. Tra un pasto e l'altro il cameriere deve dare conferma per far partire la preparazione del piatto onde evitare di portare due pietanze, che avrebbero dovuto mangiare in successione, nello stesso momento.

ESEMPIO: se viene ordinato un antipasto e un primo, il cameriere all'inizio darà l'ok solo per far preparare l'antipasto, per poi servirlo. Una volta che il cliente avrà finito, il cameriere darà l'ok per far preparare il primo.

Una volta terminato il pasto e i commensali vorranno andare via, dovranno pagare il conto. Dalla cassa verrà calcolato il conto e trascritto su una ricevuta (fiscale) per poi essere portata ai clienti. Una volta effettuato il pagamento, dalla cassa la copia del tavolo viene buttata e il tavolo liberato.

## 3.3. Sistema Proposto

Parleremo di ciò che il nostro sistema propone in termini di servizi e funzionalità offerte al cliente.

### 3.1 Panoramica

Partiamo adesso ad analizzare la realtà di interesse per informatizzare il processo.

Innanzitutto, possiamo stabilire che la struttura dovrà avere, per ogni “reparto produzione” (cucina, pizzeria) uno schermo per visualizzare gli ordini che arriveranno dalla sala, con la possibilità di interagire col sistema per un acknowledge quando il piatto verrà preparato (NON SI PARLA DI INTERFACCIA, SI TRATTA DI COME IL SISTEMA E L'UTENTE DOVRANNO INTERAGIRE).

Inoltre, ogni addetto alla sala dovrà avere un dispositivo per poter prendere le ordinazioni, in sostituzione al blocco note.

Alla cassa sarà presente un terminale con cui si potrà interagire per gestire gli ordini e monitorare lo stato della sala, evitando tutte le copie cartacee delle comande presenti nel sistema cartaceo.

Per velocizzare la comunicazione col sistema da parte degli addetti alla sala, le operazioni che non riguardano il dialogo in real-time coi clienti (ordinazione) potranno essere svolte da un terminale presente in sala così da avere una visione maggiore (tra tablet e monitor) del sistema.

Passiamo ora alla traduzione dei processi dal sistema cartaceo al sistema informatizzato

I passi da effettuare saranno, in corrispondenza:

- Dopo aver fatto accomodare i clienti e aver portato il menù da un dispositivo portatile verrà visualizzata una schermata con tutti i tavoli presenti in sala; selezionato il tavolo apparirà una nuova schermata, contenente tutti i prodotti del ristorante, divisi per categoria.
- Verranno selezionati con un click i prodotti scelti dal cliente, indicando la quantità e/o eventuali variazioni.
- Dopo aver terminato di selezionare i prodotti, verranno inviati i dati al sistema e, di conseguenza:
  - o Il resoconto dell'ordine sarà visibile nella cucina in attesa di essere preparato;
  - o Sul terminale alla cassa, il tavolo che ha effettuato l'ordinazione avrà il suo stato in occupato; (all'atto, non ci interessa qui vedere cosa ha ordinato).

Qualsiasi modifica successiva dell'ordine verrà effettuata dall'addetto alla sala. Man mano che gli ordini verranno evasi, ogni reparto aggiornerà lo stato dell'ordine in evaso, e verrà rimosso dalla schermata.

Una volta terminato il pasto e i commensali vorranno andare via, dovranno pagare il conto. Verrà avvisato il cassiere di emettere il conto, così che lui dal terminale cliccherà sul tavolo e, visualizzato il riepilogo dell'ordine, cliccherà su stampa. Una volta effettuata la stampa, il tavolo dal sistema verrà liberato.

### 3.2 3. Requisiti Funzionali

Come descritto brevemente nell'Introduzione, le operazioni che un utente può eseguire dipendono dal suo ruolo nel sistema. Nel caso di più utenti con lo stesso ruolo, gli stessi, possono eseguire le medesime operazioni su tutti i dati presenti nello scope del ruolo.

Di seguito vengono descritte in modo semplice e chiaro, quali sono le principali funzioni dei quattro differenti utenti disponibili, per la prima ed iniziale revisione del software.

#### RF 1.0 Gestione della sala

- 1.1 Visualizzare lo stato di ogni tavolo;
- 1.2 Selezionare un tavolo;
- 1.3 Stampare il conto.

#### RF 2.0 Gestione ordinazione

- 2.1 Creazione ordinazione;
- 2.2 Visualizzare un ordinazione con i relativi dettagli;
- 2.3 Visualizzare il dettaglio di ogni piatto presente nell'ordinazione selezionata;
- 2.4 Aggiungere un piatto ad un'ordinazione;
- 2.5 Rimuovere un piatto da un'ordinazione;
- 2.6 Modificare un'ordinazione;

#### RF 3.0 Gestione ristorante

- 3.1 Visualizzare lo storico degli ordini (giornaliero, settimanale, mensile o annuale);
- 3.2 Visualizzare la lista dei tavoli;
- 3.3 Visualizzare la lista dei piatti;
- 3.4 Visualizzare la lista degli ingredienti;
- 3.5 Aggiungere un nuovo tavolo;
- 3.6 Rimuovere un tavolo registrato;
- 3.7 Aggiungere un nuovo piatto al menu;
- 3.8 Rimuovere un piatto dal menu;
- 3.9 Modificare un piatto presente nel menu;

#### RF 4.0 Gestione utenti

- 4.1 Aggiungere nuovo utente al sistema;
- 4.2 Rimuovere un utente dal sistema;
- 4.3 Accesso mediante login;
- 4.4 Log out.

### 3.3 3. Requisiti non funzionali

Di seguito riportiamo i requisiti non funzionali che il sistema dovrà rispettare. I requisiti non funzionali specificano delle caratteristiche qualitative che il sistema dovrà fornire.

#### 3.3.1 Usabilità

Il sistema avrà due schermate principali: una per i terminali, e una per i dispositivi mobili per gli addetti alla sala.

L'interazione con l'utente non prevede esperienza pregressa: difatti l'utente, avrà a disposizione tutte le funzionalità in modo chiaro e leggibile.

La schermata del terminale, avrà uno schema dei tavoli in sala, colorati in base al loro stato (rosso-occupato, verde-libero) per cui riuscirà in modo chiaro a capire lo stato generale del ristorante.

Per quanto riguarda l'addetto alla sala, una volta selezionato il tavolo anch'esso dalla medesima schermata, avrà un elenco dei prodotti raggruppati, all'avvio, per categoria. In questo modo avrà a primo impatto tutte le sezioni del menù.

#### 3.3.2 Affidabilità

Più il software sarà corretto, più di conseguenza sarà affidabile.

Il sistema dovrebbe evitare la perdita dei dati in real-time dei tavoli, sia al momento dell'invio dell'ordinazione, sia, ad esempio, durante la stampa di un conto.

Il sistema dovrebbe evitare di incorrere ad errori quando si effettuano delle operazioni.

Onde prevenire problemi hardware dei terminali, si è ideato di installare due terminali: uno alla cassa, uno nella zona di appoggio dei camerieri; in questo modo sarà possibile utilizzare i terminali da due postazioni, e nel caso di malfunzionamento di uno dei due terminali sarà possibile continuare ad usufruire dell'applicazione dal terminale ancora on-line. Essendo un'applicazione Client-Server si cercherà sempre di mantenere isolato il server dalle macchine utilizzate, per evitare danni accidentali. Il sistema prevede un sistema di autenticazione e di autorizzazione per ogni utente.

#### 3.3.3 Performance

Per garantire efficienza durante l'utilizzo dell'applicazione, essendo che tutta la comunicazione dei terminali e dei dispositivi avverrà tramite rete wireless, sarà divisa la banda di utilizzo: su una frequenza di 5GHz verranno fatti viaggiare solo dati per la comunicazione del sistema, mentre la frequenza di 2.4GHz della rete verrà lasciata alla clientela.

In questo modo i tempi di utilizzo per le operazioni dovrebbero essere molto rapidi poiché, nel caso peggiore, per un inserimento di un ordine non verranno inviati più di 2MB di dati. Per il resto, non dovrebbero esserci tempi critici di risposta. In ogni evenienza, sarà accettato un ritardo di dieci secondi da parte del sistema per la ricezione dell'ordine.

#### 3.3.4 Manutenibilità

La manutenzione del sistema verrà affidata ad uno degli sviluppatori del progetto.

Manutenzione inteso anche come estensione: si è pensato, infatti, che il software potrebbe espandersi includendo anche una porzione di software per la gestione del magazzino



interno. In questo modo sarà anche possibile tenere traccia dei prodotti acquistati e venduti.

### 3.3.5 Implementazione

L'applicazione verrà sviluppata utilizzando Java come linguaggio di programmazione per la logica di business, WildFly in qualità di Web Server e MySQL, database relazionale, per la raccolta dati.

Verrà utilizzato il paradigma MVC per lo sviluppo dell'applicativo, in modo da separare i concetti della presentazione, dalla logica implementativa.

### 3.3.6 Sicurezza

Il sistema sarà utilizzato solo da utenti opportunamente loggati. Sarà l'amministratore a registrare nuovi utenti nel sistema, evitando in tal modo l'accesso da parte di utenti non autorizzati.