

|  |
| --- |
| System Design Document  Versione 1.0  **ANNO ACCADEMICO 2020/2021** |
|  |
| RISTOMANAGER  Ambrosio Salvatore  Costante Marco  Benitozzi Simone  Nappo Carla Alessia |



PARTECIPANTI

|  |  |
| --- | --- |
| **NOME** | **MATRICOLA** |
| Salvatore Ambrosio | 0512106166 |
| Costante Marco | 0512105772 |
| Benitozzi Simone | 0512105742 |
| Nappo Carla Alessia | 0512105956 |

REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **DATA** | **VERSIONE** | **DESCRIZIONE** | **AUTORE** |
| 03/12/2020 | 1.0 | Prima stesura del template | Costante Marco |
| 04/12/2020 | 1.1 | Aggiunta contenuti | Nappo Carla Alessia |

# INDICE

[INDICE 4](#_Toc57901030)

[1.Introduzione 5](#_Toc57901031)

[1.1 Scopo del sistema 5](#_Toc57901032)

[1.2 Obiettivi di Design 5](#_Toc57901033)

[1.3 Definizioni 5](#_Toc57901034)

[1.4 Acronimi 5](#_Toc57901035)

[1.5 Riferimenti 5](#_Toc57901036)

[1.6 Panoramica 5](#_Toc57901037)

[2. Architettura del software corrente 5](#_Toc57901038)

[3. Architettura del software proposto 5](#_Toc57901039)

[3.1 Panoramica 5](#_Toc57901040)

[3.2 Decomposizione del sistema 5](#_Toc57901041)

[3.3 Mapping hardware e software 5](#_Toc57901042)

[3.4 Gestione dei dati persistenti 5](#_Toc57901043)

[3.5 Controllo degli accessi e sicurezza 5](#_Toc57901044)

[3.6 Controllo del software globale 5](#_Toc57901045)

[3.7 Condizioni boundary 5](#_Toc57901046)

[**3.7.1 Avvio del sistema 5**](#_Toc57901047)

[**3.7.2 Terminazione del sistema 5**](#_Toc57901048)

[**3.7.3 Fallimento del sistema 5**](#_Toc57901049)

[4. Servizi dei sottosistemi 5](#_Toc57901050)

[5. Glossario 5](#_Toc57901051)

# 1.Introduzione

## **1.1 Scopo del sistema**

L’obiettivo del sistema è venir incontro all’esigenze dei ristoratori, che a seguito della diffusione del COVID-19 necessitano sempre di più di sistemi sicuri, nonché una rapida gestione di quelle che sono le tipiche esigenze di un’attività ristorativa.

Lo scopo, quindi, è quello di informatizzare la gestione del locale attraverso un sistema che permetta non solo di limitare al massimo i contatti col personale di sala, ma anche di tener traccia della clientela per poi poter avvisare in caso di esposizione al contagio.

Il sistema permette al cliente di accedere al menù, comporre il proprio ordine e mandare la comanda in cucina, il tutto mediante il proprio dispositivo.

RistoManager offre, inoltre, la possibilità di gestire in modo efficiente il menù, il quale soggetto a periodici cambiamenti.

## **1.2 Obiettivi di Design**

1.2.1 Criteri di performance

* Tempo di risposta: RistoManager deve essere reattivo per tutte le operazioni più immediate come l’aggiunta di un prodotto. Per la visualizzazione del menù deve garantire dei tempi di risposta brevi, ma, essendo un sistema web, dipenderà molto dalla qualità di connessione con il quale il dispositivo navigherà.
* Throughput: I picchi di carico devono essere gestiti dal sistema senza rallentamenti, garantendo fluidità.
* Memoria: Il sistema utilizza un database relazionale per memorizzare tutti i dati che non rappresenterà un problema di performance del sistema.

1.2.2 Criteri di affidabilità

* Robustezza: I componenti devono essere affidabili ed essere in grado di mantenere i propri dati anche in caso di guasti. Inoltre deve gestire eventuali input non validi da parte degli utenti.
* Disponibilità: RistoManager deve essere disponibile all’uso 24 ore su 24 da parte degli utenti grazie ad un server sempre attivo.
* Tolleranza all’errore: Il sistema deve essere capace di operare durante condizione d’errore.

1.2.3 Criteri di manutenzione

* Estendibilità: La progettazione del sistema sarà condotta in modo da agevolare la facile introduzione di nuove funzionalità.
* Modificabilità: Deve essere possibile intervenire sul codice esistente per correggere eventuali bugs o implementare nuove funzionalità. Bisogna garantire che il codice sia leggibile per rendere agevole la modifica.
* Leggibilità: Il codice sarà ben strutturato per semplificare eventuali interventi sy di esso.
* Tracciabilità dei requisiti: Sarà possibile effettuazione le modifiche necessarie al corretto funzionamento del sistema.

1.2.4 Criteri per l’utente finale

* Usabilità: RistoManager rende ogni funzione di semplice uso garantendo un’ottima esperienza all’utente che dovrà utilizzarlo grazie ad interfacce grafiche intuitive.

## **1.3 Definizioni**

* RistoManager: nome del sistema che verrà sviluppato.
* Cliente generico: utente non registrato al sistema.
* Cliente abilitato: utente registrato che può effettuazione l’ordine.
* Personale gestione: attore che gestisce il sistema.
* Personale sala: attore che genera codice.
* Personale cucina: attore che accetta e conclude gli ordini.

## **1.4 Acronimi**

# RAD: Requirements Analysis Document

# SDD: System Design Document

# DB: Database

# UC: Use Case

# SEQD: Sequence Diagram

# MU: Mock-Ups

# RF: Functional Requirements

# NRF: Non-Functional Requirements

## **1.5 Riferimenti**

* Bern Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering - Using UML, Patterns, and JAVA, 3rd edition.

## **1.6 Panoramica**

Il documento si compone di cinque parti. In particolare, nella prima parte sono stati introdotti gli obiettivi di design e sono stati forniti i riferimenti ad altri materiali. In seguito, nella sezione sistema software corrente, verrà descritto un sistema software simile. La sezione sistema software proposto documenta il system design del nuovo sistema. In questa sezione vengono descritti i seguenti elementi:

* + Decomposizione in sottosistemi: il sistema viene suddiviso in diversi sottosistemi. Ricordiamo che un sottosistema è formato da un insieme di classi, associazioni, operazioni e vincoli che sono in relazione tra di loro. Ogni sottosistema è caratterizzato dai servizi che offre agli altri sottosistemi.
  + Mapping hardware/software: in questa sezione vengono prese decisioni riguardo le piattaforme hardware su cui il sistema dovrà girare, una volta decise le piattaforme è necessario mappare le componenti su di esse.
  + Gestione dei dati persistenti: descrive i dati persistenti che vengono memorizzarli dal sistema ed il tipo di infrastruttura usata per memorizzarli.
  + Controllo degli Accessi e Sicurezza: descrive il modello degli utenti del sistema in termini di una matrice degli accessi.
  + Controllo Globale del Software: descrive il modo in cui è implementato il controllo globale del software e come si sincronizzano i sottosistemi.
  + Condizioni di boundary: vengono descritte le condizioni limite del sistema come start-up e shutdown e la gestione dei fallimenti del sistema. La sezione servizi dei sottosistemi descrive in termini di operazioni quali sono i servizi forniti da ciascun sottosistema.

L’ultima parte del documento è costituita dal glossario che si occupa di elencare una serie di termini e fornire la relativa spiegazione in maniera tale da fornire supporto a coloro che leggono il documento.

# 2. Architettura del software corrente

# 3. Architettura del software proposto

## **3.1 Panoramica**

## **3.2 Decomposizione del sistema**

## **3.3 Mapping hardware e software**

## **3.4 Gestione dei dati persistenti**

## **3.5 Controllo degli accessi e sicurezza**

## **3.6 Controllo del software globale**

## **3.7 Condizioni boundary**

### 3.7.1 Avvio del sistema

### 3.7.2 Terminazione del sistema

### 3.7.3 Fallimento del sistema

# 4. Servizi dei sottosistemi

## **4.1 Gestione Utenti**

|  |  |  |
| --- | --- | --- |
| DESCRZIONE: | Sottosistema che gestisce e raccoglie tutte le funzionalità di cui possono usufruire gli utenti del sistema. | |
| SERVIZI OFFERTI | | |
| SERVIZIO | | DESCRZIONE |
| Registrazione | | Permette al cliente di inserire le proprie generalità e il codice del tavolo per poter accedere al menù ed effettuare un ordine |
| Prenotazione tavolo | | Permette al cliente di prenotare un tavolo scegliendo data, ora e numero di posti |
| Login | | Permette allo staff di poter accedere alla propria area del sistema |
| Logout | | Permette allo staff di disconnettersi dal sistema |
| Visualizzazione dati | | Permette di visualizzare la clientela di un determinato intervallo temporale |
| Eliminazione utente | | Permette di rimuovere un membro dello staff dal sistema |
| Generazione codice | | Permette di generare un codice univoco per il tavolo |

## **4.2 Gestione Comande**

|  |  |  |
| --- | --- | --- |
| DESCRZIONE: | Sottosistema che gestisce e raccoglie le funzionalità che riguardano la gestione delle comande inviate in cucina. | |
| SERVIZI OFFERTI | | |
| SERVIZIO | | DESCRZIONE |
| Aggiungi prodotto | | Permette al cliente di aggiungere portate alla propria comanda |
| Rimuovi prodotto | | Permette al cliente di rimuovere portate dalla comanda |
| Riepilogo comanda | | Permette al cliente di visualizzare la comanda prima di inviarla ed effettuare modifiche |
| Conferma ordine | | Permette al cliente di inviare la comanda in cucina |

## **4.3 Gestione Cucina**

|  |  |  |
| --- | --- | --- |
| DESCRZIONE: | Sottosistema che gestisce e raccoglie le funzionalità che riguardano la gestione, l’esecuzione e il lavoro della cucina | |
| SERVIZI OFFERTI | | |
| SERVIZIO | | DESCRZIONE |
| Visualizzazione comanda | | Permette di visualizzare in tempo reale tutte le comande che arrivano in cucina |
| Accettazione comanda | | Permette di prendere in carico una comanda |
| Conferma comanda | | Permette di notificare il completamento della comanda |

## **4.4 Gestione Prodotti/menu**

|  |  |  |
| --- | --- | --- |
| DESCRZIONE: | Sottosistema che gestisce e raccoglie le funzionalità che riguardano la gestione, l’aggiornamento e la consultazione del menù. | |
| SERVIZI OFFERTI | | |
| SERVIZIO | | DESCRZIONE |
| Aggiungi prodotto | | Permette al gestore di aggiungere un prodotto al menù |
| Modifica prodotto | | Permette al gestore di modificare le caratteristiche di un prodotto sul menù |
| Rimuovi prodotto | | Permette al gestore di rimuovere un prodotto dal menù |
| Genera pietanza casuale | | Permette al cliente di generare una portata in modo casuale |
| Visualizza per filtro | | Permette al cliente di visualizzare le portate secondo un determinato filtro |
| Visualizza per categoria | | Permette al cliente di visualizzare i prodotti secondo una determinata categoria |

# 5. Glossario

# Client: componente che accede ai servizi o alle risorse di un'altra componente, detta server.

# Deployment Diagram: Schema che descrive la struttura dinamica del sistema

# DBMS: programma informatico (o, più frequentemente, un insieme di programmi) progettato per gestire un database, ovvero un insieme di numerosi dati strutturati. Le operazioni, normalmente, sono richieste da un gran numero di utenti.

# Form: finestra di dialogo incorporata in una pagina Web che consente all'utente di inserire informazioni destinate ad un server. Generalmente richiede un programma sul server che si occupi di esaminare le informazioni inviate. E’ composto da spazi (campi) predefiniti, ad esempio menù a tendina, elenchi puntati o caselle di testo libero.

# JDBC: API per il linguaggio di programmazione Java che serve ai client per connettersi a un database. Fornisce metodi per interrogare e modificare i dati. È orientata ai database relazionali.

# Login: Procedura attraverso la quale ci si collega con un qualsiasi servizio in linea. All'utente viene assegnato un nome di login ed una password che vengono richiesti dal sistema ogni volta che ci si collega.

# Layer: E’ un insieme di classi con funzionalità simile (tipicamente raggruppati in un unico package).

# Logout: Operazione attraverso la quale si termina un collegamento con un sistema al quale si ha accesso attraverso un nome utente e una password.

# MySQL: Database management system relazionale, composto da un client con interfaccia a caratteri e un server, disponibile su molte piattaforme.

# Password: È un metodo di sicurezza che, mediante una stringa di caratteri, permette di identificare un utente specifico. Generalmente le password sono formate da una sequenza di lettere e numeri; digitando correttamente questi caratteri, si può avere accesso al computer o alla rete.