|  |
| --- |
| **logo_standard Università degli Studi di Salerno Corso di Ingegneria del Software** |

****

**WhereDoIEat  
System Design Document  
Versione 1.2**

[1. Introduzione 3](#_Toc30442721)

[1.1. Descrizione del Problema 3](#_Toc30442722)

[1.2 Design Goals 3](#_Toc30442723)

[1.2.1 DG\_0 Criteri di Performance 3](#_Toc30442724)

[1.2.2 DG\_1 Criteri di Affidabilità 4](#_Toc30442725)

[1.2.3 DG\_2 Criteri di Costo 4](#_Toc30442726)

[1.2.4 DG\_3 Criteri di Mantenimento 4](#_Toc30442727)

[1.2.5 DG\_4 Criteri di Utente 5](#_Toc30442728)

[1.3 Definizioni, Acronimi e Abbrevazioni 5](#_Toc30442729)

[1.4 Riferimenti 5](#_Toc30442730)

[1.5 Panoramica 5](#_Toc30442731)

[2. Architettura del sistema 6](#_Toc30442732)

[2.1. Panoramica 6](#_Toc30442733)

[2.2 Decomposizione in sottosistemi 6](#_Toc30442734)

[2.2.1 Macrodecomposizione in sistemi 6](#_Toc30442735)

[2.2.2 Microdecomposizione in sottosistemi 7](#_Toc30442736)

[2.3 Mappatura Hardware/Software 11](#_Toc30442737)

[2.4 Gestione dei Dati Persistenti 12](#_Toc30442738)

[2.4.1 Class Diagram 12](#_Toc30442739)

[2.4.2 Modello Logico 12](#_Toc30442740)

[2.4.3 Struttura Tabelle 12](#_Toc30442741)

[2.5 Gestione degli accessi 15](#_Toc30442742)

[2.6 Condizione limite 17](#_Toc30442743)

[2.6.1 Start-up 17](#_Toc30442744)

[2.6.2 Start-up (a seguito di un fallimento) 17](#_Toc30442745)

[2.6.3 Terminazione 17](#_Toc30442746)

[2.6.4 Fallimento 17](#_Toc30442747)

[3. Servizi dei Sottosistemi 18](#_Toc30442748)

[3.1 SS\_UR – Gestione Account 18](#_Toc30442749)

[3.2 SS\_UR – Gestione Autenticazione 18](#_Toc30442750)

[3.3 SS\_UR – Gestione Prenotazione 18](#_Toc30442751)

[3.4 SS\_UR – Gestione Recensioni 19](#_Toc30442752)

[3.5 SS\_UR – Gestione Visualizzazione 19](#_Toc30442753)

[3.6 SS\_UR – Gestione Lista Preferiti 20](#_Toc30442754)

[3.7 SS\_RIS – Gestione Attività 20](#_Toc30442755)

[3.8 SS\_UR – Gestione Segnalazioni 21](#_Toc30442756)

[3.9 SS\_RIS – Gestione Amministrazione 21](#_Toc30442757)

[4. Glossario 22](#_Toc30442758)

# 1. Introduzione

## 1.1. Descrizione del Problema

Nell’epoca moderna è diventato di fondamentale importanza il mondo digitale, inteso come mezzo utile per comunicare ma anche per pubblicizzare le proprie attività ristorative e non.

Proprio per questo esistono vari sistemi, come per esempio Trip Advisor, che appunto si occupano di pubblicizzare le attività, fornire un criterio di valutazione per i clienti che hanno intenzione di provare quest’ultime e in generale di darle maggiore visibilità sul territorio.

Questi sistemi però non sono in grado di gestire la prenotazione, un cliente quando visita un’attività che magari lo stuzzica, è impossibilitato a prenotare un tavolo, dovrà provvedere a tale operazione magari chiamando direttamente l’attività.

Per ovviare a questo problema, appunto, è nato il sistema “WhereDoIEat”, il quale, oltre a permettere la valutazione da parte dei clienti dell’attività con le recensioni, esso permette anche di prenotare un tavolo in quell’attività (se l’attività ne dispone).

## 1.2 Design Goals

La Web App WhereDoIEat punterà ad essere il più lineare ed intuitiva possibile. Per Far ciò si avvarrà di una struttura grafica chiara e completa, con bottoni, finestre di dialogo e icone. Cercherà inoltre di isolare le informazioni necessarie per indirizzare più facilmente l’utente verso la funzionalità da lui richiesta. Valore aggiunto all’applicazione sarà la sua semplicità, che permetterà anche agli utenti con scarsa conoscenza del sistema di portare a termine le loro operazione, evitando di inserire dati sbagliati o commettete errori durante l’utilizzo dell’applicazione.

Il sistema proposto rispetterà i seguenti criteri di design:

### 1.2.1 DG\_0 Criteri di Performance

* DG\_0.1 Tempi di risposta: Il sistema deve garantire tempi di risposta brevi per ogni funzionalità. Mediamente una richiesta dovrà essere soddisfatta in un tempo non superiore ai 5 secondi. Ovviamente quest’ultimo può oscillare in base alla velocità di connessione.
* DG\_0.2 Throughput: Il sistema sarà capace di gestire una media di 200 utenti, consentendo loro di effettuare tutte le operazioni senza subire alcun rallentamento. L’applicazione sarà anche in grado di gestire un maggiore afflusso di utenti qualora sia necessario.
* DG\_0.3 Dati persistenti: Il sistema, per memorizzare tutti i dati relativi agli utenti e alle foto utilizzerà un database relazionale. La scelta di quest’ultimo farà in modo che le informazioni saranno il più velocemente possibile.

### 1.2.2 DG\_1 Criteri di Affidabilità

* DG\_1.1 Robustezza: Il sistema mostrerà un messaggio che avvertirà l’utente nel caso in cui i dati siano mancanti o errati. Questo consentirà di non immettere dati sbagliati all’interno del database.
* DG\_1.2 Affidabilità: Il sistema garantirà il corretto svolgimento di tutte le funzionalità e cercherà di produrre l’output atteso evitando errori indesiderati.
* DG\_1.3 Disponibilità: Il sistema dovrà essere sempre fruibile agli utenti. Non sono previsti periodi di chiusura del sistema se non quelli per motivi di manutenzioni.
* DG\_1.4 Sicurezza: Il sistema, tramite username e password, riuscirà ad individuare il tipo di utente e gli permetterà di effettuare solo le operazioni appartenenti alla sua categoria.

### 1.2.3 DG\_2 Criteri di Costo

* DG\_2.1 Costo: Prima dell’inizio dello sviluppo della piattaforma verranno valutati i costi di sviluppo del sistema, costi di manutenzione e costi di amministrazione.

### 1.2.4 DG\_3 Criteri di Mantenimento

* DG\_3.1 Modificabilità: il sistema permetterà di apportare modifiche alle funzionalità già implementate senza la necessità di modificare i sottosistemi.

### 1.2.5 DG\_4 Criteri di Utente

* DG\_4.1 Utilità: Il sistema punterà a soddisfare tutte le esigenze degli utenti sviluppando tutti i requisiti funzionali in fase di analisi.
* DG\_4.2 Usabilità: Il sistema dovrà essere chiaro ed intuitivo e anche l’utente con meno esperienza dovrà essere in grado di effettuare le operazioni desiderate. La piattaforma consentirà “l’autoapprendimento” cioè l’utente effettuerà le operazioni in maniera sempre più rapida con il maggiore utilizzo del sistema.

## 1.3 Definizioni, Acronimi e Abbrevazioni

|  |  |
| --- | --- |
| Acronimo | Descrizione |
| RAD | Requirement Analysis Document |
| JDBC | Java DataBase Connectivity |
| DBMS | Database Management System |
| HTTP | HyperText Transfer Protocol |
| GUI | Graphic User Interface |
| DB | DataBase |

## 1.4 Riferimenti

* RAD WhereDoIEat
* Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition
* <https://it.wikipedia.org/>

## 1.5 Panoramica

Il documento si compone di 3 parti fondamentali:

1. Introduzione: In questa sezione verranno illustrati gli obiettivi del sistema proposto e saranno descritte, in modo non approfondito, le varie funzionalità messe a disposizione per gli attori. Verranno poi mostrati gli obiettivi del design, in particolar modo i criteri che il sistema dovrà rispettare.
2. Architettura del sistema proposto: In questa sezione verrà elaborata la macrocomposizione in sottosistemi, la gestione dei dati persistenti, il mapping hardware/software del sistema, il controllo degli accessi , sicurezza e le condizioni limite.
3. Servizi dei sottosistemi e Glossario: In questa sezione troveremo la descrizione per ogni servizio fornito del sottosistema e un glossario contenente una raccolta dei termini contenuti nel sistema proposto.

# 2. Architettura del sistema

## 2.1. Panoramica

Il sistema prevederà l’utilizzo di un architettura ibrida tra Client- Server e Model-View-Controller.

Il Lato Client si occuperà delle pagine di presentazione del sistema(View)

Il Lato Server si occuperà della logica applicativa e di controllo (rispettivamente Model e Controller).

## 2.2 Decomposizione in sottosistemi

### 2.2.1 Macrodecomposizione in sistemi

La macro decomposizione divide il sistema nei seguenti sottosistemi:

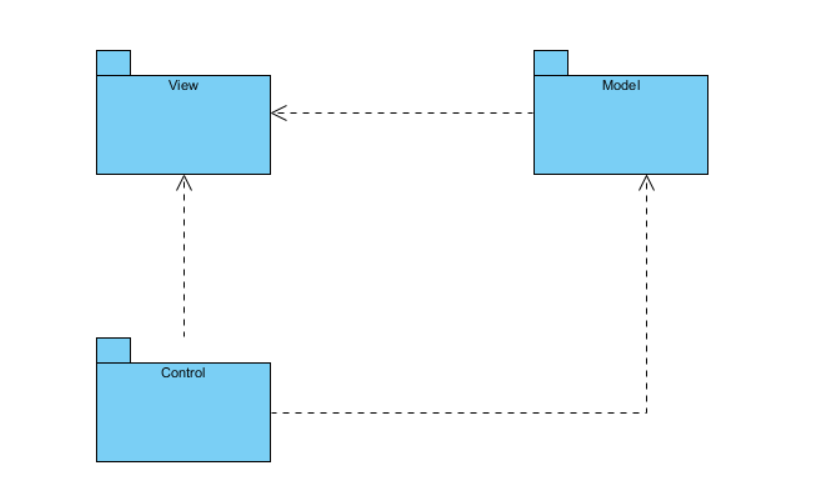
1. Gestione Account.
2. Gestione Autenticazione
3. Gestione Prenotazioni.
4. Gestione Recensioni.
5. Gestione Visualizzazioni.
6. Gestione Lista preferiti.
7. Gestione Attività.
8. Gestione Segnalazioni.
9. Gestione dell’amministratore.

Gli utenti che useranno il sistema lo faranno dal proprio computer comunicando gli input all’interfaccia del Server Web i quali verranno gestiti dal Database in cui sono contenute tutte le informazioni dell’intero sistema. Il Database sarà gestito da un DBMS che si occupa di inserire, cercare e aggiornare i dati presenti al suo interno, elaborando la richiesta degli utenti da parte del Server. Il DBMS si occuperà anche di gestire gli accessi concorrenti al Database.

### 2.2.2 Microdecomposizione in sottosistemi

Per semplificare la progettazione e lo sviluppo dell’applicazione, i sottosistemi saranno decomposti secondo lo schema previsto dall’architettura software MVC, ovvero:

1. View: si occupa di gestire l’interfaccia grafica del sottosistema e contiene tutti gli elementi che interagiscono con l’utente.
2. Control: si occupa, dati gli input dagli utenti, di elaborarli e fornire dei risultati che verranno poi mostrati graficamente.
3. Model: è la parte del sottosistema che interagisce col Database.



#### 2.2.2.1. Gestione Account

Questo sottosistema si occupa della gestione dell’account, in particolare fornisce le funzioni di:

* Modifica della password
* Visuaizzazione area personale
* Modifica dati account

#### 2.2.2.2. Gestione Autenticazione

Questo sottosistema si occupa della gestione dell’autenticazione all’interno del sistema, in particolare permette di:

* Registrarsi al sistema
* Effettuare il login da parte di un utente già registrato
* Effettuare il logout

#### 2.2.2.3. Gestione Prenotazione

Questo sottosistema si occupa della gestione della prenotazione, in particolare fornisce le funzioni di:

* Prenotazione di un locale
* Annulla prenotazione

#### 2.2.2.4. Gestione Recensioni

Questo sottosistema si occupa della gestione delle recensioni, in particolare fornisce le funzioni di:

* Valutazione e recensione di un locale
* Modifica di una valutazione già effettuata
* Modifica di una recensione gia effettuata
* Visualizzazione valutazioni e recensioni effettuate da parte di un Cliente
* Lettura recensioni di un’attività da parte del ristoratore
* Risposta alle recensioni ricevuta da parte di un ristoratore

#### 2.2.2.5. Gestione visualizzazione

Questo sottosistema si occupa della gestione delle visualizzazioni, in particolare fornisce le funzioni di:

* Visualizzazione tramite barra di ricerca
* Visualizzazione tramite home
* Visualizzazione tramite categorie
* Visualizzazione tramite comune
* Visualizzazione recensioni delle attività
* Visualizzazione mappa del locale

#### 2.2.2.6. Gestione Lista Preferiti

Questo sottosistema si occupa della gestione della lista preferiti ,in particolare permette ad utente registrato di:

* Aggiungere un attività alle sue preferite
* Visualizzare le sue attività preferite
* Eliminazione di un’attività dai preferiti

#### 2.2.2.7. Gestione Attività

Questo sottosistema si occupa della gestione delle attività, in particolare fornisce le funzioni di:

* Aggiunta attività al sistema
* Eliminazione attività dal sistema
* Visualizzazione proprie attività
* Visualizzazione dati specifici di una singola attività
* Modifica posizione locale
* Inserimento nuove foto del locale
* Eliminazione foto del locale

#### 2.2.2.8. Gestione Segnalazioni

Questo sottosistema si occupa della gestione delle segnalazioni, in particolare fornisce le funzioni di:

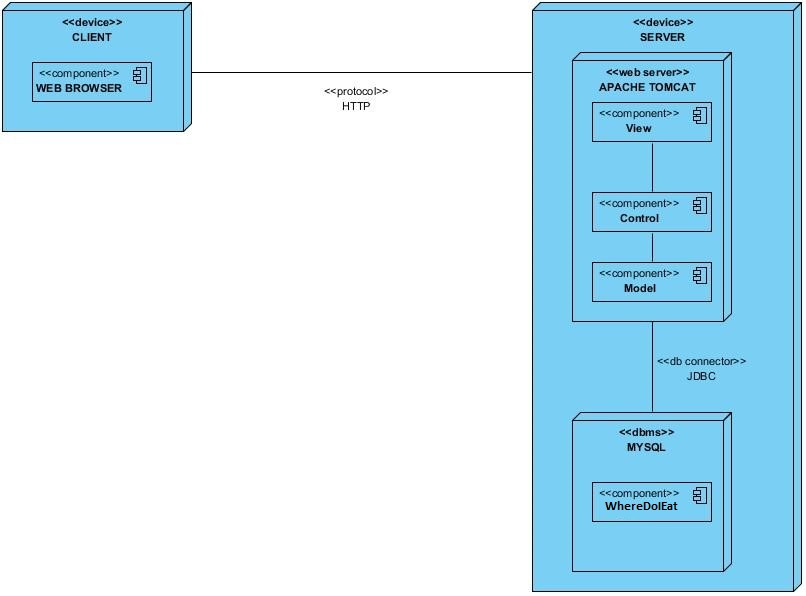
* Visualizzazione suggerimenti effettuati
* Amministrazione suggerimenti
* Effettua Suggerimento

#### 2.2.2.9. Gestione Amministrazione

Questo sottosistema si occupa della gestione dell’amministrazione dell’intero sistema, in particolare fornisce le funzioni di:

* Visualizzazione dati di ogni utente registrato
* Visualizzazione utenti registrati
* Eliminazione account
* Eliminazione commenti non appropriati
* Modificare password utenti

## 2.3 Mappatura Hardware/Software

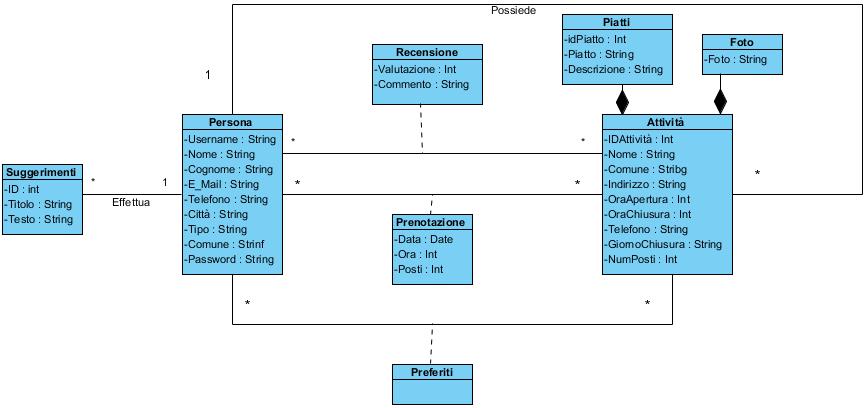
****

Il sistema utilizza un’architettura client/server, in cui un server fornisce servizi a più client. Su una macchina client è eseguito un browser web che consente all’utente di interagire a livello View(lato server) per inoltrare richieste e visualizzare le risposte ricevute. La macchina server gestisce la logica applicativa e i dati persistenti. La comunicazione tra client e server avviene tramite protocollo http. Questo protocollo permette di trasferire ipertesti, consentendo a due macchine, client e server, di interagire attraverso un meccanismo di richiesta e risposta. Il client inoltra una richiesta al server che verrà soddisfatta con la risposta di quest’ultimo. Per il client, le specifiche hardware sono una qualsiasi macchina dotata di connessione a internet, mentre per il software, un sistema operativo con un web browser installato. Per il server, invece, le specifiche hardware consistono di una macchina connessa a internet capace di immagazzinare dati a sufficienza. Le specifiche software necessarie comprendono un Database Management System (MySQL) per la gestione dei dati persistenti, un Web Server (Apache Tomcat) per la gestione della logica applicativa e della comunicazione con più client.

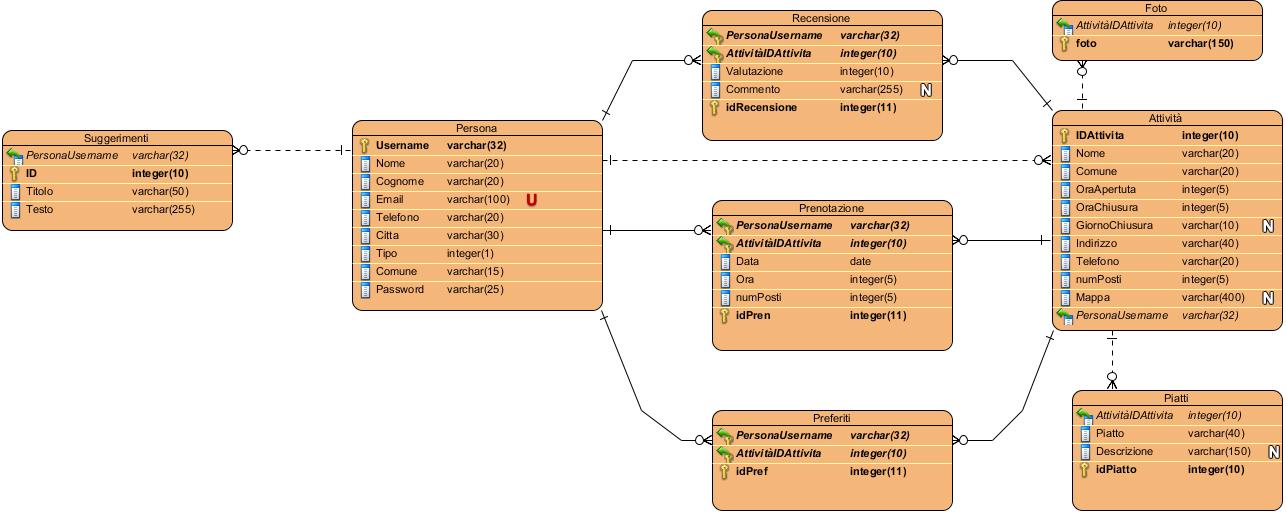
## 2.4 Gestione dei Dati Persistenti

Per la gestione dei dati persistenti, essi saranno memorizzati in un database relazionale in quanto le informazioni da memorizzare sono strutturate e complesse. Il sistema WhereDoIEat utilizzerà un DBMS capace di effettuare le operazioni principali tra cui inserimento, rimozione e visualizzazione dei dati immagazzinate nel sistema. Per questo abbiamo scelto MySQL, che soddisfa tutti i nostri requisiti.

### 2.4.1 Class Diagram



### 2.4.2 Modello Logico

****

### 2.4.3 Struttura Tabelle

**Tabella: Persona**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| username | VARCHAR(32) | Primaria |
| nome | VARCHAR(20) |  |
| cognome | VARCHAR(20) |  |
| Email | VARCHAR(100) |  |
| Telefono | VARCHAR(20) |  |
| Citta | VARCHAR(30) |  |
| Tipo | INTEGER(1) |  |
| Comune | VARCHAR(15) |  |
| Password | VARCHAR(25) |  |

**Tabella: Attività**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| IDAttivita | INTEGER(10) | Primaria |
| Nome | VARCHAR(20) |  |
| Comune | VARCHAR(20) |  |
| Email | VARCHAR(100) |  |
| OraApertura | INTEGER(5) |  |
| OraChiusura | INTEGER(5) |  |
| GiornoChiusura | VARCHAR(10) |  |
| Indirizzo | VARCHAR(40) |  |
| Telefono | VARCHAR(25) |  |
| NumPosti | INTEGER(5) |  |
| Mappa | VARCHAR(60) |  |
| PersonaUsername | VARCHAR(32) | Chiave esterna |

**Tabella: Suggerimenti**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| ID | INTEGER(10) | Primaria |
| Titolo | VARCHAR(50) |  |
| Testo | VARCHAR(255) |  |
| PersonaUsername | VARCHAR(32) | Chiave esterna |

**Tabella: Recensione**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| Valutazione | INTEGER(10) |  |
| Commento | VARCHAR(255) |  |
| PersonaUsername | VARCHAR(32) | Chiave esterna |
| AttivitàIDAttivita | INTEGER(10) | Chiave esterna |

**Tabella: Prenotazione**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| Data | DATE |  |
| Ora | INTEGER(5) |  |
| NumPosti | INTEGER(5) |  |
| AttivitàIDAttivita | INTEGER(10) | Chiave esterna |
| PersonaUsername | VARCHAR(32) | Chiave esterna |

**Tabella: Preferiti**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| AttivitàIDAttivita | INTEGER(10) | Chiave esterna |
| PersonaUsername | VARCHAR(32) | Chiave esterna |

**Tabella: Piatti**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| idPiatto | INTEGER(10) | Chiave primaria |
| Piatto | VARCHAR(40) |  |
| Descrizione | VARCHAR(150) |  |
| AttivitàIDAttivita | INTEGER(10) | Chiave esterna |

**Tabella: Foto**

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Chiave** |
| Foto | VARCHAR(150) | Chiave primaria |
| AttivitàIDAttivita | INTEGER(10) | Chiave esterna |

## 2.5 Gestione degli accessi

Essendo WhereDoIEat un sistema con differenti tipi di utenza, essi possono accedere a diverse funzionalità, a seconda di quale oggetto hanno intenzione di interagire.

Per documentare i diritti d’accesso per ogni attore quindi, ecco rappresentata una matrice che suddivide la tipologia di attore per colonna, e la tipologia di oggetto a cui si accede per riga, per ogni combinazione (Attore, Oggetto) è presente l’insieme delle operazioni disponibili.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attori** | **Cliente** | **Amministratore** | **Ristoratore** |
| **Oggetti** |
| -Persona | -Registrazione()  -Login()  -Logout()  -VisualizzaAreaPersonale()  -ModificaPassword()  -ModificaDati() | -Login()  -Logout()  -DeleteUtente()  -EditPass()  -VisualizzaDati()  -GetAllUtenti() | --Registrazione()  -Login()  -Logout()  -VisualizzaAreaPersonale()  -ModificaPassword()  -ModificaDati() |
| -Suggerimenti | -FaiSuggerimento()  -VediSuggerimenti() | -GetSuggerimentiBug() | -FaiSuggerimento() |
| -Attività | -RicercaLocale()  -RicercaLocali()  -VisualizzazioneDati()  -GetMappa() | -VisualizzaDati() | -NewAttività()  -EliminaAttività()  -VisualizzaDati()  -GetAllAttività()  -ModificaPosizione() |
| -Recensione | -GetRecensione()  -VediRecensioni()  -FaiRecensione()  -ModificaValutazione()  -ModificaRecensione() | -VediRecensioni()  -DeleteRecensione() | -VediRecensioni()  -FaiRisposta() |
| -Prenotazione | -Prenota()  -AnnullaPrenotazione()  -VisualizzaPrenotazioni() |  | VisualizzaPrenotazioni() |
| -Preferiti | -VisualizzaAttivitàPreferite()  -GetInfoAttivitàPreferita()  -AddPreferiti()  -DeleteAttivitàPreferita() |  |  |
| -Foto | VisualizzaFoto() |  | -AggiungiFoto()  -EliminaFoto() |
| -Piatti | VisualizzaPiatti() |  | -AggiungiPiatto()  -EliminaPiatto() |

## 2.6 Condizione limite

### 2.6.1 Start-up

Per il primo start-up del sistema è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l’interpretazione ed esecuzione del codice lato server. In seguito, verrà mostrata l’home page della piattaforma dalla quale è possibile entrare nell’area della Login, dove sarà possibile autenticarsi tramite opportune credenziali (username o e-mail e password) come utente con pieno accesso a tutte le funzionalità del sistema.  
Una volta effettuato l'accesso, sarà possibile effettuare tutte le operazioni consentite, che saranno diverse in base al tipo dell’utente.

### 2.6.2 Start-up (a seguito di un fallimento)

Il sistema può subire guasti dovuti al sovraccarico del database con successivo fallimento. Per ovviare al problema, periodicamente è previsto un salvataggio dei dati sotto forma di codice SQL. All’avvio a seguito di tale fallimento, oltre alle normali procedure previste per lo start-up, l’ultimo codice SQL memorizzato sarà eseguito per la rigenerazione del database.

### 2.6.3 Terminazione

Al momento della chiusura dell’applicativo si ha la terminazione del sistema con un regolare Logout dal sistema. Viene assicurata la consistenza dei dati, annullando eventuali operazione che erano in esecuzione.

### 2.6.4 Fallimento

Possono verificarsi diversi casi di fallimento del sistema:

Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso, è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.

Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, non sono previsti metodi che ripristinino lo stato del sistema a uno stato antecedente allo spegnimento inaspettato.

Un altro caso di fallimento potrebbe derivare dal software stesso che causa un crash inaspettato dovuto ad errori commessi durante la fase di implementazione, e non sono previste politiche correttive, l’unico processo che potrà essere eseguito è la chiusura del sistema e il suo successivo riavvio.

# 3. Servizi dei Sottosistemi

## 3.1 SS\_UR – Gestione Account

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Account | Gestisce le operazioni riguardanti la gestione dell’area personale e la modifica dei dati personali. |
| **Servizio** | **Descrizione** |
| ModificaPassword() | Permette all’utente di poter effettuare la modifica della propria password. |
| VisualizzaDati() | Permette all’utente di visualizzare la propria area personale e di visualizzare i propri dati. |
| ModificaDati() | Permette all’utente di modificare i propri dati personali. |

## 3.2 SS\_UR – Gestione Autenticazione

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Autenticazione | Gestisce le operazioni riguardanti l’autenticazione per permettere ad un utente di registrarsi al sistema o di effettuare login/logout. |
| **Servizio** | **Descrizione** |
| Registrazione() | Permette ad un utente non registrato di potersi registrare al sistema. |
| Login() | Permette all’utente di poter accedere al sistema ed usufrire le funzionalità di un utente registrato. |
| Logout() | Permette all’utente di potersi disconettere dal sistema. |

## 3.3 SS\_UR – Gestione Prenotazione

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Prenotazione | Si occupa delle operazioni riguardanti la gestione delle prenotazioni. |
| **Servizio** | **Descrizione** |
| PrenotaLocale() | Permette all’utente registrato di poter effettuare una prenotazione ad un locale. |
| AnnullaPrenotazione() | Permette all’utente registrato di poter annullare una prenotazione effettuata in precedenza |

## 3.4 SS\_UR – Gestione Recensioni

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Recensioni | Gestisce le operazioni riguardanti le recensioni, permette ad un utente di effettuare, modificare o visualizzare recensioni di un’attività. |
| **Servizio** | **Descrizione** |
| FaiValERec() | Permette all’utente di poter aggiungere una nuova recensione ad una attività. |
| ModificaVal() | Permette all’utente di poter modificare la valutazione di una recensione già effettuata riguardante un’attività presente sul sistema. |
| ModificaRec() | Permette all’utente di poter modificare il commento(lasciando invariata la valutazione) di una recensione già effettuata riguardante un’attività presente sul sistema. |
| VediValERecFatte() | Permette all’utente di poter visualizzare le recensioni(composte dalla valutazione più il commento) rilasciate alle attività. |
| VediRecensioniRicevute() | Permette al ristoratore di visualizzare le recensioni effettuate ad una delle sue attività. |
| RispondiRecensione() | Permette al ristoratore di poter rispondere ad una delle recensioni ricevute da uno dei suoi clienti. |

## 3.5 SS\_UR – Gestione Visualizzazione

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Visualizzazione | Si occupa delle operazioni riguardanti la gestione della visualizzazione delle attività e dei dati di quest’ultime. |
| **Servizio** | **Descrizione** |
| VisualizzaAttività() | Permette all’utente di visualizzare le attività tramite il tipo di ricerca effettuata. |
| VisualizzaRecensione() | Permette all’utente di visualizzare le recensioni effettuate da altri utenti su un’attività. |
| VisualizzaMappa() | Permette all’utente di visualizzare la mappa di un’attività |

## 3.6 SS\_UR – Gestione Lista Preferiti

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Lista Preferiti | Si occupa delle operazioni riguardanti la gestione della lista preferiti, permettendo ad un utente registrato al sistema di poter usufruire della lista preferiti delle attività. |
| **Servizio** | **Descrizione** |
| AddToFavorite() | Permette all’utente registrato di poter aggiungere un’ attività nella sua lista preferiti. |
| SeeAllFavorite() | Permette all’utente registrato di poter visualizzare la sua lista delle attività preferite. |
| DeleteFromFavorite() | Permette all’utente registrato di poter rimuovere una delle attività dalla lista preferiti. |

## 3.7 SS\_RIS – Gestione Attività

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Attività | Si occupa delle operazioni riguardanti la gestione, l’inserimento e l’eliminazione di attività consentita agli utenti che posseggono un’attività. |
| **Servizio** | **Descrizione** |
| AggiungiAttività() | Permette all’utente di poter aggiungere una nuova attività al sistema. |
| EliminaAttività() | Permette all’utente di poter rimuovere una sua attività dal sistema. |
| VisualizzaAttività() | Permette all’utente di visualizzare tutte le sue attività presenti sul sistema. |
| VisualizzaSingolaAttività() | Permette all’utente di visualizzare i dettagli specifici della sua attività che ha selezionato. |
| ModificaPosizione() | Permette all’utente di modificare la posizione di una sua attività che ha selezionato. |
| InserisciFoto() | Permette all’utente di poter aggiungere una foto ad una sua attività presente sul sistema. |
| EliminaFoto() | Permette all’utente di poter rimuovere una foto presente su una delle sue attività. |

## 3.8 SS\_UR – Gestione Segnalazioni

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Segnalazioni | Si occupa delle operazioni riguardanti i Suggerimenti effettuati dagli utenti. |
| **Servizio** | **Descrizione** |
| VisualizzaSuggerimenti() | Permette all’utente di visualizzare i suggerimenti che ha inviato all’amministratore. |
| AmministraSuggerimenti() | Permette all’amministratore di leggere i suggerimenti effettuati dagli utenti. |
| EffettuaSuggerimento() | Permette all’utente di effettuare un nuovo suggerimento. |

## 3.9 SS\_RIS – Gestione Amministrazione

|  |  |
| --- | --- |
| **Sottosistema** | **Descrizione** |
| Gestione Amministrazione | Si occupa delle operazioni riguardanti la gestione e l’amministrazione degli utenti e dell’intero sistema. |
| **Servizio** | **Descrizione** |
| Login() | Permette all’Amministratore di poter accedere al sistema ed usufruire le funzionalità di un amministratore. |
| Logout() | Permette all’utente di potersi disconnettere dal sistema. |
| EliminazioneCommenti() | Permette all’amministratore di eliminare una recensione da un’attività. |
| ModificaPassword() | Permette all’amministratore di modificare la password degli utenti. |
| VisualizzaDati() | Permette all’amministratore di visualizzare i dati dell’utente. |
| VisualizzaUtenti() | Permette all’amministratore di visualizzare la lista degli utenti registrati. |
| EliminaAccount() | Permette all’amministratore di eliminare un utente. |

# 4. Glossario

|  |  |
| --- | --- |
| ***Nome*** | ***Significato*** |
| ***WhereDoIEat*** | *Nome del Sistema che verrà sviluppato.* |
| ***Utente non registrato*** | *Il termine indica la tipologia di utente non registrata al sistema, e dispone delle funzionalità base del sistema(come visualizzazione di attività e dei dati di quest’ultime).* |
| ***Amministratore*** | *Il termine indica una particolare tipologia di operatore che si occupa anche dell’amministrazione e della gestione di tutti gli operatori del sistema.* |
| ***Utente registrato*** | *Il termine indica la tipologia di utente registrato al sistema, disponde delle funzionalità di base ed in più può effettuare recensioni alle attività, aggiungere alla lista preferiti, fare seganalazioni o effettuare una prenotazione.* |
| ***Ristoratore*** | *Il termine indica la tipologia di utente registrato al sistema, però si occupa delle funzionalità inerenti alla gestione delle attività.* |
| ***Suggerimenti*** | *Il termine indica un messaggio che un utente vuole inviare a un amministratore del sistema, specificando i problemi che ha avuto.* |
| ***Recensione*** | *Il termine identifica una valutazione numerica da 1 a 5 ed un commento, che vengono effettuati da un utente registrato ad una determinata attività, per fornire un feedback su quest’ultima.* |
| ***Prenotazione*** | *Il termine identifica un utente registrato che sta tentando di prenotare un locale, fornendo la ora, la data e il numero dei posti.* |
| ***Preferiti*** | *Il termine identifica una lista contenente tutte le attività che vengono salvate da un utente registrato, per avere un accesso più facile e rapido a quest’ultime.* |
| ***Attività*** | *Il termine identifica un locale presente sul sistema, la quale è possibile consultare le sue informazioni, ed eventualmente aggiungerlo alla lista preferiti o effettuare una prenotazione.* |
| ***Foto*** | *Il termine indica l’indirizzo logico usato per indicare la posizione della foto.* |
| ***Piatti*** | *Il termine identifica le informazioni(descrizione,nome) inerenti ad uno dei piatti tipici dell’attività.* |