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

****

**WhereDoIEat  
System Design Document  
Versione 1.0**

[**1. Introduzione**](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_tqboc5hf8lh8)5

[1.1 Descrizione](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_1fob9te) del Problema 5

1.2 Design Goals 6

1.2.1 DG\_0 Criteri di Performance 6

1.2.2 DG\_1 Criteri di Affidabilità 7

1.2.3 DG\_2 Criteri di Costo 7

1.2.4 DG\_3 Criteri di Mantenimento 7

1.2.5 DG\_4 Criteri di Utente 8

1.3 Definizioni,Acronimi e Abbreviazioni 8

1.4 Riferimenti 8

1.5 Panoramica 8

[**2.**](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_t5j8pddtg2l2) **Architettura del Sistema** 9

[2.1](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_c392ex7wtyt) Panoramica 9

[2.2](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_ye1pg675k47i) Decomposizione in sottosistemi 9

2.2.1 Macrodecomposizione in sottosistemi 9

2.2.2 Microdecomposizione in sottosistemi 10

2.2.2.1 Gestione Account 11

2.2.2.2 Gestione VenditaFoto 13

2.2.2.3 Gestione Abbonamento 15

2.2.2.4 Gestione Carta di Credito 16

2.2.2.5 Gestione Statistiche 17

2.2.2.6 Gestione Operatori 19

2.2.2.7 Gestione Segnalazioni 21

2.2.2.8 Gestione Assistenza 23

2.2.2.9 Gestione Interazioni 25

2.2.2.10 Gestione Autenticazione 26

2.2.2.11 Gestione Post 28

2.2.2.12 Gestione FotoAcquistate 31

2.3 Mappatura Hardware/Software 32

2.4 Gestione dei Dati Persistenti 33

2.4.1 Modello E-R 33

2.4.2 Modello E-R Ristrutturato 34

2.4.3 Modello Logico 23

2.4.4 Struttura Tabelle 24

2.5 Gestione degli accessi 25

2.6 Condizione limite 26

2.6.1 Start up 26

2.6.2 Start-up (a seguito di un fallimento) 26

2.6.3 Terminazione 26

2.6.4 Fallimento 27

[**3. Servizi**](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_ouyp94swbe30) **dei Sottosistemi 21**

[3.1​ ​](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_bb57r5ky8ysx)Gestione Operatori 21

[3.2 ​](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_wqijn9dwk2tb)Gestione Assistenza 22

[3.3](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_9fd8v0l6clph) Gestione Segnalazioni 22

[3.4](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_msmvm06z1j7d) Gestione Account 23

[3.5](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_tr5gtjq1ppyy) Gestione Segnalazioni 23

[3.6](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_l8lb8ff5yg5a) Gestione Statistiche 23

[3.7](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_4b76vy2p8llt) Gestione Abbonamento 24

[3.8 Gestione](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_4b76vy2p8llt) Carta di Credito 24

3.9 Gestione Interazioni 25

3.10 Gestione Autenticazione 27

3.11 Gestione Post 28

3.12 Gestione FotoAcquistate 31

[**4.**](file:///C:\Users\ilida\Desktop\APPOGGIO\gilberto%20e%20CIMINO\gilb\SDD_V.1.1.docx#_6tmevnpmt5ii) **Glossario** 2**6**

**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. **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. **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. **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. **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. **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. **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. **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. **Riferimenti**
* RAD WhereDoIEat
* Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition
* <https://it.wikipedia.org/>
  1. **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).

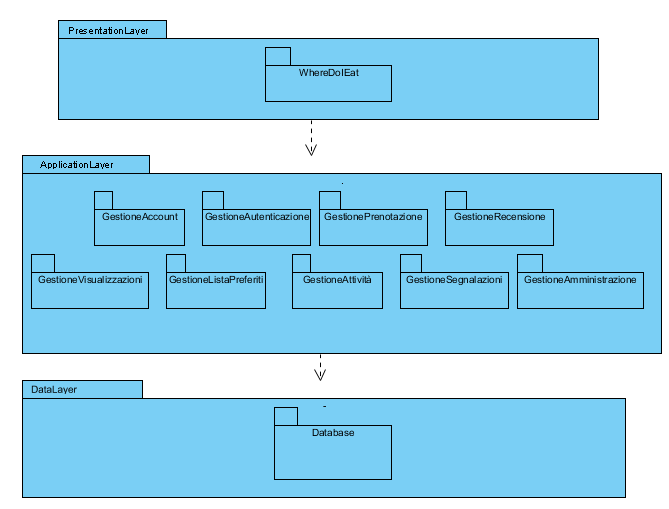
## **2.2 Decomposizione in sottosistemi**

### **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.

Di seguito è mostrato il diagramma in cui vengono evidenziate le relazioni di dipendenza tra i package di ogni sottosistema e tra i diversi sottosistemi.

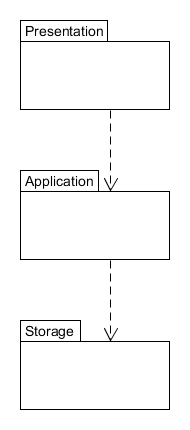


Gli utenti che useranno il sistema lo faranno dal proprio computer comunicando gli input all’interfaccia del Server Web i quali verranno gestiti dal Database in cui sono contenute tutte le informazioni dell’intero sistema. Il Database sarà gestito da un DBMS che si occupa di inserire, cercare e aggiornare 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 Three-Layer, ovvero:

1. Presentation: si occupa di gestire l’interfaccia grafica del sottosistema e contiene tutti gli elementi che interagiscono con l’utente.
2. Application: si occupa, dati gli input dagli utenti, di elaborarli e fornire dei risultati che verranno poi mostrati graficamente.
3. Storage: 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

Immagine che contiene screenshot

Descrizione generata automaticamente

**GestioneAccountPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUIVisualizzaDati: comprende le interfacce che consentono all’Utente Registrato di visualizzare le informazioni relative al proprio account.
* GUIModificaDati: comprende le interfacce che consentono all’Utente Registrato di modificare le informazioni relative al proprio account.
* GUIModificaPassword: comprende le interfacce che consentono all’Utente Registrato di modificare la password relativa al proprio account

**GestioneAccountApplicationLayer**

* ModificaPassword(): operazioni per modificare la propria password.
* VisualizzaDati(): operazioni per la visualizzazione delle informazioni relative al proprio account.
* ModificaDati(): operazioni per la modifica delle informazioni al proprio account.i

**GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi all’account.

**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

**Immagine che contiene screenshot

Descrizione generata automaticamente**

**GestioneAccountPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUIAggiuntaAttività: comprende le interfacce che consentono al ristoratore di aggiungere un’attività all’interno del sistema.
* GUIEliminazioneAttività: comprende le interfacce che consentono al ristoratore di eliminare un’attività dal sistema.
* GUIProprieAttività: comprende le interfacce che consentono al ristoratore di visualizzare dati generali sulle proprie attività.
* GUISingolaAttività: comprende le interfacce che consentono al ristoratore di visualizzare dati specifici di una propria singola attività.
* GUIModificaPosizione: comprende le interfacce che consentono al ristoratore di modificare la posizione di una propria attività.
* GUIInserimentoFoto: comprende le interfacce che consentono al ristoratore di inserire foto per una propria attività.
* GUIEliminazioneFoto: comprende le interfacce che consentono al ristoratore di eliminare foto da una propria attività.

**GestioneAccountApplicationLayer**

* aggiungiAttività(): operazione per aggiungere un’attività all’interno del sistema.
* eliminaAttività(): operazione per eliminare un’attività dal sistema.
* visualizzaAttività(): operazione per visualizzare dati generali sulle proprie attività.
* visualizzaSingolaAttività: operazione per visualizzare dati specifici di una propria singola attività.
* modificaPosizione(): operazione per modificare la posizione di una propria attività.
* inserisciFoto(): operazione per inserire foto per una propria attività.
* eliminaFoto(): operazione per eliminare foto da una propria attività.

**GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle attività

**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

**Immagine che contiene testo

Descrizione generata automaticamente**

**GestioneAccountPresentationLayer**

Include tutti gli elementi dell’interfaccia grafica che offrono funzionalità riguardanti la gestione dell’account. Comprende:

* GUIValERec: comprende le interfacce che consentono al Cliente dell’attività di rilasciare una recensione cioè un commento ed una valutazione.
* GUIModificaVal: comprende le interfacce che consentono al Cliente dell’attività di modificare una valutazione già effettuata.
* GUIModificaRec: comprende le interfacce che consentono al Cliente dell’attività di modificare una recensione già effettuata.
* GUIValERecFatte: comprende le interfacce che consentono al Cliente dell’attività di vedere le recensioni e la valutazioni già fatte.
* GUIRecensioniRicevute: comprende le interfacce che consentono al ristoratore di vedere le recensioni ricevute per una propria attività.
* GUIRispondiRecensione: comprende le interfacce che consentono al ristoratore di rispondere a delle recensioni ricevute su una propria attività.

**GestioneAccountApplicationLayer**

* faiValERec(): operazione per fare una valutazione ed un commento.
* modificaVal(): operazione per modificare una valutazione fatta.
* modificaRec(): operazione per modificare un commento fatto.
* vediValERecFatte(): operazione per far si che un Cliente possa visualizzare le recensioni fatte alle attività fin’ora.
* vediRecensioniRicevute(): oprazione per far si che un ristoratore possa vedere le recensiomni ricevute su una propria attività.
* rispondiRecensione(): operazione per far si che il ristoratore possa rispondere ad una recensione ricevuta su una sua attività.

**GestioneAccountDataLayer**

Si occupa di rendere reperibili i dati, presenti all’interno del database, relativi alle recensioni.