Università degli studi di Salerno

Dipartimento di Informatica

***Corso di Laurea in Informatica***



***INGEGNERIA DEL SOFTWARE***

***Problem Statement***

***“VORWERK”***

*.*

**Docente:**

Andrea De Lucia

**Studenti:**

##### Nome Matricola

Vincenzo Marrazzo 0512105832

*Anno Accademico: 2020/21*

INDICE

Sommario

[**1.** **INTRODUZIONE** 3](#_Toc60569799)

[1.1 Scopo del sistema 3](#_Toc60569800)

[1.2 Design Goals 3](#_Toc60569801)

[1.2.1 Criteri di performance (Performance Criteria) 4](#_Toc60569802)

[1.2.2 Criteri di affidabilità (Dependability Criteria) 4](#_Toc60569803)

[1.2.3 Criteri di mantenimento (Maintenance Criteria) 4](#_Toc60569804)

[1.2.4 Utente finale (End User Criteria) 4](#_Toc60569805)

[1.3 Definizioni, Acronimi ed Abbreviazioni 5](#_Toc60569806)

[1.4 Riferimenti 5](#_Toc60569807)

[1.5 Visione Generale 5](#_Toc60569808)

[**2.** **SISTEMA SOFTWARE CORRENTE** 6](#_Toc60569809)

[**3.** **SISTEMA SOFTWARE PROPOSTO** 6](#_Toc60569810)

[3.1 Visione Generale 6](#_Toc60569811)

[3.2 Decomposizione in sottosistemi 7](#_Toc60569812)

[3.2.1 Decomposizione in Layer 7](#_Toc60569813)

[3.2.2 Decomposizione in Sottosistemi 8](#_Toc60569814)

[3.3 Mapping hardware/software 11](#_Toc60569815)

[3.4 Controllo Accessi e Sicurezza 12](#_Toc60569816)

[3.5 Controllo flusso globale del sistema 13](#_Toc60569817)

[3.6 Condizioni limite 13](#_Toc60569818)

[3.7 Start-up 14](#_Toc60569819)

[3.7.1 Terminazione 14](#_Toc60569820)

[3.7.2 Fallimento 14](#_Toc60569821)

[**4.** **Servizi dei Sottosistemi** 15](#_Toc60569822)

[4.1 Applicazione Layer 15](#_Toc60569823)

[4.1.1 Gestione Account 15](#_Toc60569824)

[4.1.2 Gestione Robot 15](#_Toc60569825)

[4.1.3 Gestione Acquisto Robot 16](#_Toc60569826)

[4.1.4 Gestione Ricezione Premio 16](#_Toc60569827)

[4.1.5 Gestione Carrello 17](#_Toc60569828)

[4.1.6 Gestione Premi 17](#_Toc60569829)

[**5.** **Glossario** 18](#_Toc60569830)

# **INTRODUZIONE**

## Scopo del sistema

Con il seguente progetto mi appresto ad analizzare e gestire l’insieme delle problematiche relative all’amministrazione di un e-commerce. Tale attività è suddivisa in tre aree fondamentali, la prima e la seconda area riguardano l’amministrazione dei prodotti, cioè robot e premi, e la terza area riguarda l’interazione tra il cliente e il sistema.

L’obbiettivo generale del progetto VORWERK è, dunque, quello di implementare un sistema software semplice da utilizzare e che permetta l’acquisto di un robot da cucina e del ricevimento di un premio senza recarsi nei negozi abilitati.

## Design Goals

Abbiamo diverse categorie di design goals:

* Criteri di performance
* Criteri di affidabilità
* Criteri di costo
* Criteri di manutenzione
* Criteri utenti finali

### Criteri di performance (Performance Criteria)

**Tempo di Risposta:** A qualsiasi sollecitazione fornita dagli attorei, il sistema deve rispondere entro 30 secondo oppure in caso di errore presentare un messaggio di errore.

**Throughput:** il sistema deve essere in grado di gestire molte richiesta e allo stesso tempo offrire le stesse performance

**Memoria:** La dimensione complessiva dipende dalla memoria utilizzata per il mantenimento del database. Verrà utilizzato il programma MySql; in presenza di grandi quantità di dati, non vi sarà un rallentamento del sistema.

### Criteri di affidabilità (Dependability Criteria)

**Robustezza:** Il sistema, nelle interazioni con gli attori, effettuerà dei controlli sui dati in input allo scopo di validarli. Se la validazione darà esito positivo i dati saranno successivamente elaborati; altrimenti verrà mostrato un messaggio di errore.

**Tolleranza ai guasti:** In caso di errore, il sistema non continuerà ad operare, ma mostrerà una notifica all’utente, ed eventualmente istruzioni su come proseguire. Inoltre i dati verranno salvati per evitare delle perdite.

**Sicurezza:** Nessun utente può vedere dati relativi ad altri utenti. Inoltre l’accesso avviene tramite un’e-mail, una per ogni utente, e password. Inoltre i dati sono resi sicuri, in quanto l’accesso al database è limitato solo ad alcuni operatori.

**Affidabilità:** Il sistema deve attenersi alle specifiche funzionali che sono state descritte nell’apposita documentazione (RAD)

### Criteri di mantenimento (Maintenance Criteria)

**Tracciabilità dei requisiti:** La tracciabilità dei requisiti è possibile grazie ad una matrice di tracciabilità, attraverso la quale è possibile retrocedere al requisito associato ad ogni parte del progetto. La tracciabilità è garantita dalla fase di progettazione fino al testing.

**Portabilità:** Il sistema sarà portabile in quanto l’interazione avviene mediante un browser quindi non c’è dipendenza dovuta al sistema operativo.

### Utente finale (End User Criteria)

**Usabilità:** Il sito sarà di facile utilizzo e User Friendly. Inoltre vi è un manuale utente che fornirà un supposto cartaceo agli amministratore del sistema.

**Utilità:** Il sistema risponderà in modo esaustivo alle richiesta degli utenti, che verranno supportati dal sistema.

## Definizioni, Acronimi ed Abbreviazioni

**RAD** =Requirement Analysis Document.

**RNF** = Requisito Non Funzionale.

**SDD** = System Design Document.

**MySql** = è il più diffuso database Open Source basato sul linguaggio SQL.

**VORWERK** = il sistema di gestione dell’E-commerce sviluppato.

**User-friendly** = software di facile utilizzo anche per non è esperto.

**UML** = è un metodo per descrivere l’architettura di un sistema in dettaglio.

**File** = contenitore di informazioni/dati in formato digitale.

## Riferimenti

* Prentice Object Oriented Software Engineering Using UML Patterns and Java

## Visione Generale

Nei successivi paragrafi del documento di System Design verranno descritti:

* **Sistema Software Corrente:** rappresenta l’architettura del software che è stato sostituito o, se esso non esiste, descrive l’architettura dei software per sistemi simili esistenti.
* **Sistema Software proposto:** documenta il modello di System Design del nuovo sistema in cui vengono descritti i seguenti elementi:
  + Decomposizione in sottosistemi, nella quale viene descritta la suddivisione del sistema in vari sottosistemi.
  + Hardware/Software mapping.
  + Gestione Dati Persistenti.
  + Controllo degli Accessi e Sicurezza.
  + Controllo Globale del Software.
  + Condizioni Boundary in cui sono descritte le condizioni limite del sistema.
* **Servizi dei Sottosistemi:** descrive i servizi forniti da ciascun sottosistema in termini di operazioni.

# **SISTEMA SOFTWARE CORRENTE**

Ci sono molti sistemi software che si occupano di gestire le richieste di un e-commerce tramite un sistema digitalizzato. Ma il mio scopo è quello di creare un software che sia facile da utilizzare, dove chiunque può accedervi in completa sicurezza e affidabilità.  
Si tratta, perciò, di un sistema che rientra nel campo della Re-Engineering.  
Attraverso una Re-Engineering un progetto software viene sviluppato avendo come riferimento un sistema esistente, ma utilizzato nuove tecnologie.

# **SISTEMA SOFTWARE PROPOSTO**

## Visione Generale

Il sistema che si vuole proporre è una applicazione web il cui obiettivo è fornire un supposto a tutte le figure che vengono coinvolte nella fase di acquisto di un robot da cucina oppure del ricevimento di un premio online.  
Il sistema fornirà al cliente una lista di tutti i robot e premi disponibili.  
Un sistema di gestione del progetto VORWERK è la funzionalità principale del sistema, ciò permetterà al cliente di evitare di recarsi in un’agenzia per l’acquisto di un robot o il ricevimento di un premio, questo garantisce un minor spreco di tempo in quanti il cliente può fare tutto da casa. Per le restanti figure coinvolte (Gestore Robot, Gestore Premi) viene garantita la gestione e la visibilità necessaria al sistema.  
I gestori possono gestire il sistema in quanto:  
Gestore robot può gestire i robot, ed i Gestore premi possono gestire i premi presenti entrambi nel database.

Il pattern architetturale scelto è **Client-Server,** cioè il sottosistema Server fornisce servizi ad una serie di altri sottosistemi detti Client. Il Client conosce l’interfaccia del Server(fondamentale per poter usufruire dei suoi servizi) mentre il Server non ha bisogno di conoscere l’interfaccia dei Client(funge solo da provider)

Le funzionalità saranno, infine, divise in layer logici distinti tra di loro in base al loro tipo: sarà presente il **layer di presentazione**, il **Layer di Applicazione** ed il **Layer di memorizzazione(Data Storage Layer)**.

## Decomposizione in sottosistemi

### Decomposizione in Layer

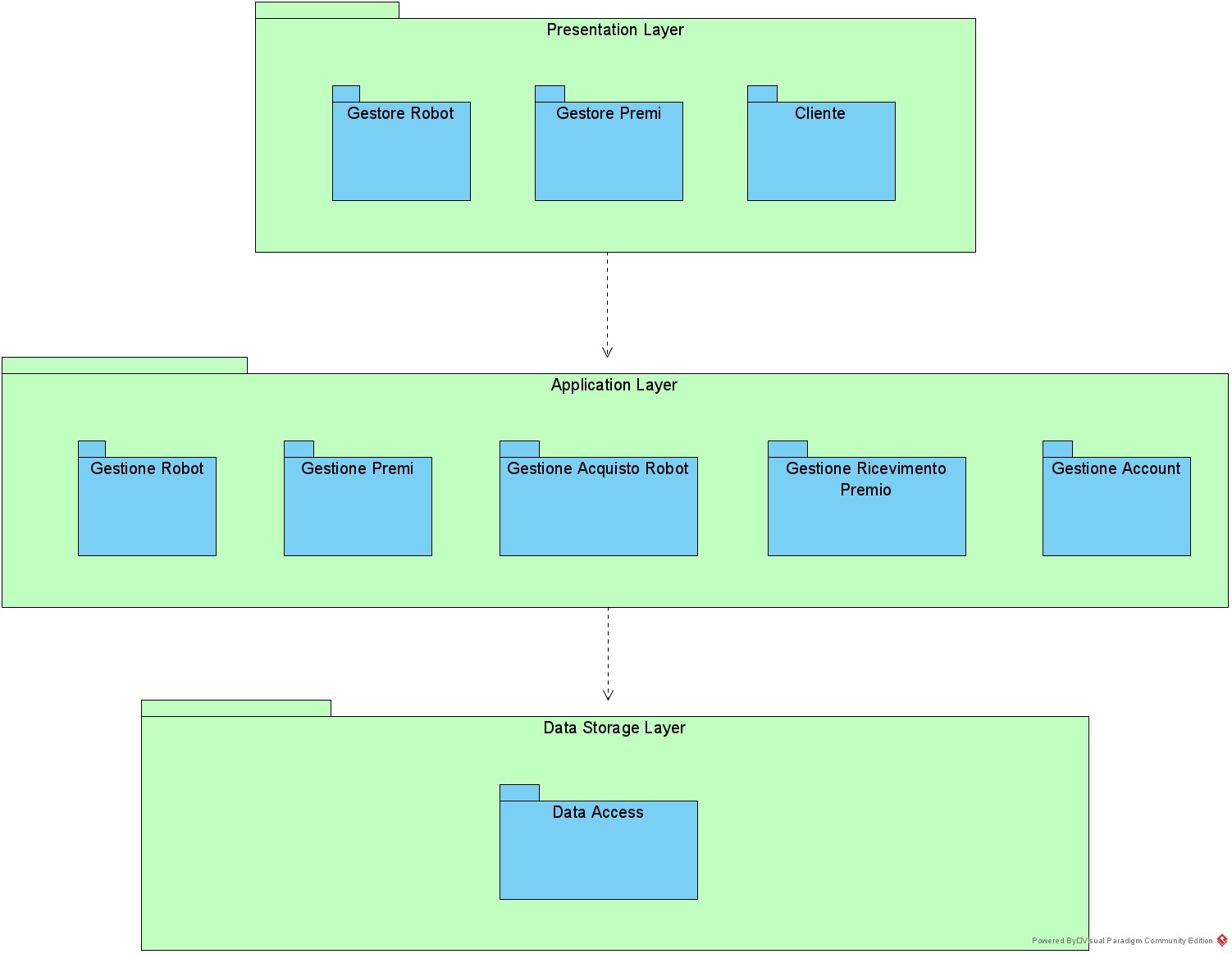
La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestirne aspetti e funzionalità differenti:

* Presentation Layer: raccoglie e gestisce l’interfaccia grafica e gli eventi generali dall’utente;
* Application Layer: si occupa della gestione della logica applicatica del sistema e fornisce servizi richiesti dal Presentation Layer;
* Data Storage Layer: si occupa della gestione dei dati persistenti e comunica con l’Application Layer fornendo i dati richiesti;

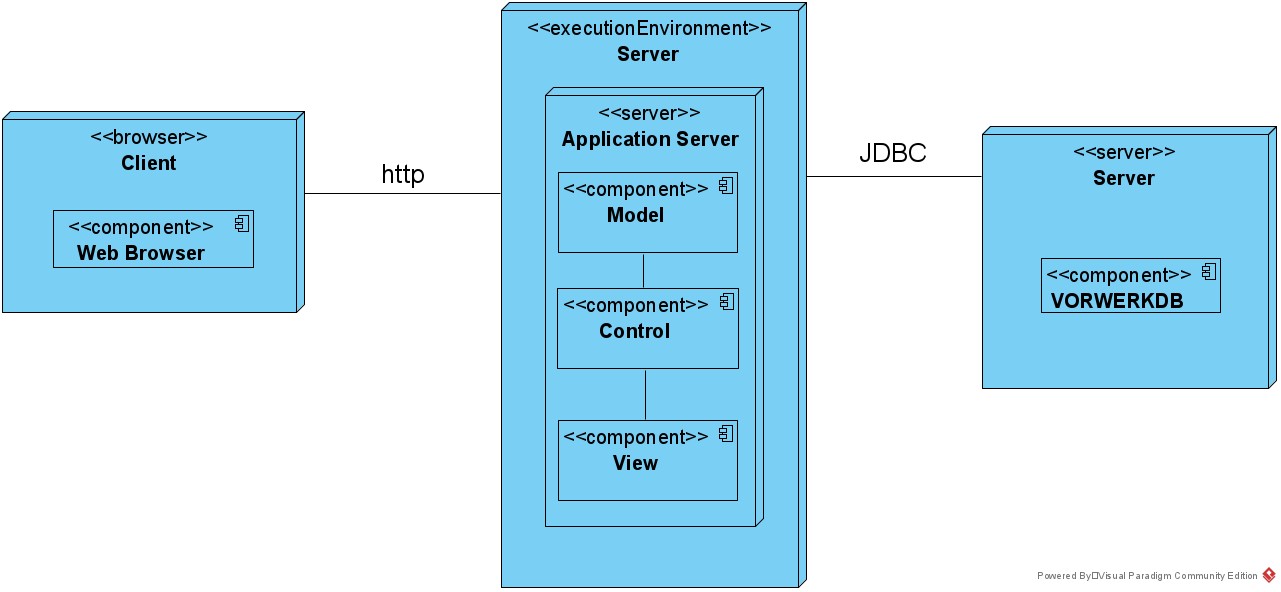
### Decomposizione in Sottosistemi

Il sistema è stato diviso in tre strati:

* Il livello di **Presentation Layer** è stato diviso in tre sottosistemi principali:
  + Le partizioni fanno riferimento ai tre principali attori che accedono al sistema:
    - Sottosistema **Cliente**: include tutte le interfaccia grafiche a cui il cliente può accedere, ovvero: registrazione, login, logout, visualizza catalogo robot, visualizza catalogo premi, ricerca robot, acquisto robot, ricevi premio, visualizza scheda tecnica robot, visualizza scheda tecnica premio, visualizza area privata, visualizza ordini, modifica dati personali.
    - Sottosistema **Gestore robot**: Include tutte le interfacce grafiche a cui il gestore robot può accedere, ovvero: login, logout, inserimento/modifica/eliminazione robot, visualizza catalogo robot.
    - Sottosistema **Gestore premi**: include tutte le interfacce grafiche a cui il gestore premi può accedere, ovvero: login, logout, inserimento/modifica/eliminazione premio, visualizza catalogo premi.
* Il livello di **Application Layer** è stato diviso in 4 sottosistemi principali.
  + Questi 4 sottosistemi rappresentano le quattro funzionalità base che svolgono gli attori:
    - Sottosistema **Gestore robot**: in esso incluse tutte le operazioni che permettono di modificare, inserire o visualizzare i robot.
    - Sottosistema **Gestore premi**: in esso incluse tutte le operazioni che permettono di modificare, inserire o visualizzare i premi.
    - Sottosistema **Gestione Acquisto Robot**: in esso sono incluse tutte le operazioni che permettono di acquistare un robot.
    - Sottosistema **Gestione Ricevimento Premio**: in esso sono incluse tutte le operazioni che permettono di riceve un premio.
    - Sottosistema **Gestione Account**: in esso sono incluse tutte le operazioni che permettono di modificare i dati personali.
* Il livello di **Data** **Storage Layer**: sistema che gestisce ed immagazzina i dati persistenti. Questo è l’unico sottosistema che deve cambiare se cambia l’interfaccia del sottosistema DATABASE.



## Mapping hardware/software



Per il sistema che si vuole sviluppare è stata scelta un’architettura Client/Server, dove ognuno dei due nodi contiene specifiche componenti.  
Tramite una qualunque device dotata di browser, l’utente può accedere all’interfaccia grafica del sistema.  
Le componenti di Business Logic, dello Storage ed il database sono istanziate nel Server.  
Questo consente di centralizzare le risorse, dato che il server è al centro della rete. In generale la comunicazione tra client e server avverrà sfruttando il protocollo http.  
L’Hyper Text Transfer Protocol (HTTP) (protocollo di trasferimento di un ipertesto) è un protocollo a livello applicativo usato come principale sistema per la trasmissione d’informazioni sul web ovvero in un’architettura tipica Client/Sever.

## Controllo Accessi e Sicurezza

Il controllo degli accessi è garantito dall’utilizzo di email e password per tutti gli utenti; tali dati verranno richiesti per ogni singolo accesso. La sicurezza è garantita, in quanto tutti i dati personali dei singoli utenti sono protetti dall’accesso controllato. Nel sistema vi sono diversi tipo di utenti che hanno in comune le seguenti funzionalità:

* **Login**
* **Logout**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sottosistema Attore | Gestione Account | | Gestione Acquisto Robot | Gestione Carrello | Gestione Ricezione Premio | Gestione Robot | Gestione Premi |
| CLIENTE | * Inserisci/ Modifica/ Elimina  Indirizzo * Inserisci/ Modifica/ Elimina Carta di credito * Visualizza Ordini * Login * Logout * Registrazione * Visualizza Profilo | * Ricerca Robot * Aggiungi/rimuovi Robot al/dal carrello * Visualizza Scheda tecnica Robot * Visualizza Robot | | * Elimina Robot dal carrello * Visualizza Robot nel Carrello * Procedi con l’acquisto dei prodotti nel carrello | * Ricevi Premio * Visualizza Premi * Visualizza Scheda tecnica Premio |  |  |
| Gestore Robot | * Login * Logout | |  |  |  | * Inserisci Robot * Modifica Robot * Elimina Robot * Visualizza Robot |  |
| Gestore Premi | * Login * Logout | |  |  |  |  | * Inserisci Premio * Modifica Premio * Elimina Premio * Visualizza Premi |

## Controllo flusso globale del sistema

Il flusso di controllo globale è la sequenza di azioni nel sistema. Il sistema VORWERK ha un flusso guidato di eventi; le funzionalità richiedono un’interazione continua da parte dell’utente; per questo motivo, il controllo del flusso globale che utilizziamo è di tipo event-driven.  
Quindi non abbiamo una sequenza di operazioni prestabilite, ma è l’utente che sceglie l’operazione da eseguire.

## Condizioni limite

Nel sistema vi sono degli oggetti persistenti; per ognuno di essi bisogna definire le fasi di creazione, eliminazione e di archiviazione nel DB.

## Gestione dei Dati Persistenti

La gestione dei dati persistenti avverrà attraverso il DBSM MySql.  
Queste sono le classi che verranno rese persistenti:

* **Utente**
* **Robot**
* **Premio**
* **PremioInOrdine**
* **Ordine**
* **Indirizzo**
* **Carrello**
* **Carta di credito**

## Start-up

Per lo start-up iniziale del sistema VORWERK, è necessario l’avvio di un web server che fornisca il servizio di un Database MySql per gestire i dati persistenti e l’interpretazione e l’esecuzione del codice lato server.

### Fallimento

I casi di fallimento del sistema possono essere diversi:

1. Guasti dovuti al sovraccarico e successivo fallimento del database. In questo caso, come provvedimento si ha quello del salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione nel DB.
2. Interruzione inaspettata dell’alimentazione, non sono previsti metodi di ripristino dello stato del sistema prima dello spegnimento.
3. Software che causa la chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. In questo caso non sono previste politiche di correzione; l’unico processo eseguibile sarà la chiusura del sistema e il suo riavvio.
4. Errore critico nell’hardware. In questo caso non sono previste soluzioni.

# **Servizi dei Sottosistemi**

## Applicazione Layer

### Gestione Account

|  |  |
| --- | --- |
| Sottosistema: Gestione Account | |
| Descrizione | Sottosistema appartenente alla stratificazione *Application Layer,* contiene un’interfaccia che gestisce l’accesso al sistema. |

|  |  |
| --- | --- |
| Servizi del sottosistema | |
| Operazioni | **Comportamento** |

|  |  |
| --- | --- |
| **Login()** | Operazione che offre la possibilità ad un utente Registrato di accedere alla sezione ad egli riservata |
| **Logout()** | Operazione che offre la possibilità ad un utente loggato di uscire dal sistema |
| **Registrazione()** | Operazione che offre la possibilità ad un Cliente di registrarsi al sistema |
| **Aggiungi/Modifica/Elimina Indirizzo()** | Operazione che offre la possibilità ad un Cliente di aggiungere/modificare/eliminare un Indirizzo |
| **Aggiungi/Modifica/Elimina CartaDiCredito()** | Operazione che offre la possibilità ad un Cliente di aggiungere/modificare/eliminare una Carta di Credito |

### Gestione Robot

|  |  |
| --- | --- |
| Sottosistema: Gestione Robot | |
| Descrizione | Sottosistema appartenente alla stratificazione *Application Layer,* contiene un’interfaccia che gestisce i robot, con funzionalità dedicate al Gestore Robot. |

|  |  |
| --- | --- |
| Servizi del sottosistema | |
| Operazioni | **Comportamento** |

|  |  |
| --- | --- |
| **AggiungiRobot()** | Operazione che offre la possibilità ad un Gestore Robot di aggiungere un Robot al sistema |
| **ModificaRobot()** | Operazione che offre la possibilità ad un Gestore Robot di modificare un Robot nel sistema. |
| **EliminaRobot()** | Operazione che offre la possibilità ad un Gestore Robot di eliminare un robot |
| **VisualizzaRobot()** | Operazione che offre la possibilità ad un Gestore Robot di visualizzare i Robot presenti nel sistema. |

### Gestione Acquisto Robot

|  |  |
| --- | --- |
| Sottosistema: Gestione Acquisto Robot | |
| Descrizione | Sottosistema appartenente alla stratificazione *Application Layer,* contiene un’interfaccia che gestisce i robot da parte del Cliente. |

|  |  |
| --- | --- |
| Servizi del sottosistema | |
| Operazioni | **Comportamento** |

|  |  |
| --- | --- |
| **RicercaRobot()** | Operazione che offre la possibilità ad un Cliente, ma anche visitatore, di ricercare un robot. |
| **VisualizzaRobot()** | Operazione che offre la possibilità ad un Cliente/Visitatore di visualizzare tutti i robot disponibili. |
| **VisualizzaSchedaTecnicaRobot()** | Operazione che offre la possibilità ad un Cliente/Visitatore di visualizzare la scheda tecnica di un robot. |
| **Aggiungi/Rimuovi Robot al/dal Carrello()** | Operazione che offre la possibilità ad un Cliente di aggiungere/rimuovere un Robot al/dal Carrello |

### Gestione Ricezione Premio

|  |  |
| --- | --- |
| Sottosistema: Gestione Ricezione Premio | |
| Descrizione | Sottosistema appartenente alla stratificazione *Application Layer,* contiene un’interfaccia che gestisce i premi da parte del Cliente. |

|  |  |
| --- | --- |
| Servizi del sottosistema | |
| Operazioni | **Comportamento** |

|  |  |
| --- | --- |
| **VisualizzaPremi()** | Operazione che offre la possibilità ad un Cliente/Visitatore di visualizzare tutti i premi disponibili. |
| **VisualizzaSchedaTecnicaPremio()** | Operazione che offre la possibilità ad un Cliente/Visitatore di visualizzare la scheda tecnica di un premio. |
| **RiceviPremio()** | Operazione che offre la possibilità ad un Cliente di ricevere un premio. |

### Gestione Carrello

|  |  |
| --- | --- |
| Sottosistema: Gestione Carrello | |
| Descrizione | Sottosistema appartenente alla stratificazione *Application Layer,* contiene un’interfaccia che gestisce il Carrello del Cliente. |

|  |  |
| --- | --- |
| Servizi del sottosistema | |
| Operazioni | **Comportamento** |

|  |  |
| --- | --- |
| **EliminaRobotDalCarrello()** | Operazione che offre la possibilità ad un Cliente di eliminare un Robot dal Carrello. |
| **VisualizzaRobotNelCarrello()** | Operazione che offre la possibilità ad un Cliente di visualizzare i Robot presenti nel Carrello. |
| **ProcediConAcquisto()** | Operazione che offre la possibilità ad un Cliente di procedere con l’acquisto dei Robot presenti nel Carrello. |

### Gestione Premi

|  |  |
| --- | --- |
| Sottosistema: Gestione Premi | |
| Descrizione | Sottosistema appartenente alla stratificazione *Application Layer,* contiene un’interfaccia che gestisce i robot, con funzionalità dedicate al Gestore Premi. |

|  |  |
| --- | --- |
| Servizi del sottosistema | |
| Operazioni | **Comportamento** |

|  |  |
| --- | --- |
| **AggiungiPremio()** | Operazione che offre la possibilità ad un Gestore Premi di aggiungere un Premio al sistema |
| **ModificaPremio()** | Operazione che offre la possibilità ad un Gestore Premi di modificare un Premio nel sistema. |
| **EliminaPremio()** | Operazione che offre la possibilità ad un Gestore Premi di eliminare un Premio. |
| **VisualizzaPremi()** | Operazione che offre la possibilità ad un Gestore Premi di visualizzare i Premi presenti nel sistema. |

# **Glossario**

**VORWERK=** Nome del Sistema.  
**RAD=** Requirements Analysis Document.  
**RNF=** Requisito Non Funzionale.  
**SDD=** System Design Document.  
**MySql=** è il più diffuso database Open Source basato sul linguaggio SQL.