Immagine che contiene testo, logo, emblema, simbolo

Descrizione generata automaticamente

Corso di Laurea in Informatica

Progetto Ingegneria del Software

OBJECT DESIGN

VERSIONE 1.1

Immagine che contiene cerchio, Elementi grafici, simbolo, Policromia

Descrizione generata automaticamente

**GameChanger**

# Partecipanti

Isidoro D’Arco – 0512113779

Ruben Petina - 0512114733

Sommario

[Partecipanti 1](#_Toc184222124)

[1. Introduzione 3](#_Toc184222125)

[2. Relazione con altri documenti 3](#_Toc184222126)

[3. Funzionalità 4](#_Toc184222127)

[4. Criteri di successo/insuccesso 4](#_Toc184222128)

[5. Approccio 4](#_Toc184222129)

[5.1 Testing di unità 4](#_Toc184222130)

[5.2 Testing di integrazione 5](#_Toc184222131)

[5.3 Testing di integrazione 5](#_Toc184222132)

[6. Sospensione e ripresa 5](#_Toc184222133)

[6.1 Criteri di sospensione 5](#_Toc184222134)

[6.2 Criteri di ripresa 5](#_Toc184222135)

[7. Materiali per il test 5](#_Toc184222136)

[8. Casi di test 6](#_Toc184222137)

[8.1 Gestione account 6](#_Toc184222138)

[8.2 Gestione account 9](#_Toc184222139)

## Introduzione

Il documento di Analisi dei Requisiti e il System Design Document hanno fornito una visione complessiva del sistema e dei suoi obiettivi, evitando di entrare nei dettagli tecnici legati all'implementazione. Questo documento ha l’obiettivo di sviluppare un modello capace di integrare tutte le funzionalità precedentemente individuate in modo preciso e coerente.

### - Object design trade-off

* **Interfaccia vs Usabilità**  
  La progettazione della GUI si è focalizzata sul garantire un’esperienza utente semplice e intuitiva. Per raggiungere questo obiettivo, sono stati adottati colori e icone facilmente riconoscibili, insieme a un design che semplifica le interazioni con il sistema. Queste scelte hanno permesso di ottenere un'interfaccia chiara, efficace e accessibile a un ampio pubblico.
* **Sicurezza vs Efficienza**  
  Pur essendo la sicurezza un elemento prioritario, i tempi limitati di sviluppo hanno portato a concentrarsi su interventi mirati. In particolare, sono stati introdotti controlli avanzati sui campi di testo per garantire l'integrità del sistema, impedendo l'inserimento di dati potenzialmente dannosi per il codice. Questa misura riduce i rischi legati a vulnerabilità come l’injection e migliora complessivamente la protezione del sistema.
* **Prestazioni vs Costi**  
  A causa delle ristrettezze di budget, si è deciso di evitare l’utilizzo di librerie e template a pagamento. Il progetto si è quindi basato sull’adozione di soluzioni open source e sull’utilizzo di risorse hardware già disponibili.

### 1.2 - Linee guida per la documentazione delle interfacce

Gli sviluppatori devono rispettare le seguenti convenzioni per la scrittura del codice:

#### Naming Convention

La nomenclatura deve essere:

* Descrittiva
* Pronunciabile
* Di lunghezza medio-corta
* Di uso comune
* Composta solo da caratteri consentiti (a-z, A-Z, 0-9)

#### Variabili

* I nomi delle variabili devono iniziare con una lettera minuscola
* Se il nome è composto da più parole, utilizzare la Lower Camel Notation (es: nomeVariabile) oppure l’underscore (\_) come separatore
* Ogni riga deve contenere la dichiarazione di una sola variabile

#### Metodi

* I nomi dei metodi devono iniziare con una lettera minuscola, con eventuali parole successive in maiuscolo (Lower Camel Notation)
* Il nome del metodo deve iniziare con un verbo che descrive l’azione, seguito da un sostantivo
* I metodi accessori e modificatori devono seguire i pattern getNomeVariabile e setNomeVariabile

#### Classi

* I nomi delle classi devono essere scritti in Pascal Case Notation, iniziando con una lettera maiuscola e con ogni parola successiva in maiuscolo (es: ProdottoMagazzino)
* I nomi delle classi devono riflettere le informazioni e le funzionalità fornite dalla classe stessa
* La struttura delle classe deve seguire questo ordine:
  + Dichiarazione della classe pubblica
  + Dichiarazione di costanti
  + Dichiarazioni di variabili di classe
  + Dichiarazioni di variabili di istanza
  + Costruttore
  + Metodi

#### Packages

* I nomi dei packages dovranno essere scritti in minuscolo concatenando insieme diversi sostantivi o sigle, separate dal carattere “.”

### - Definizione, acronimi e abbreviazioni

**RAD** sta per Requirement Analysis Document

**SDD** sta per System Design Document

**ODD** sta per Object Design Document

**JSP** sta per Java Servlet Page

### – Riferimenti

Riferimento al RAD e SDD