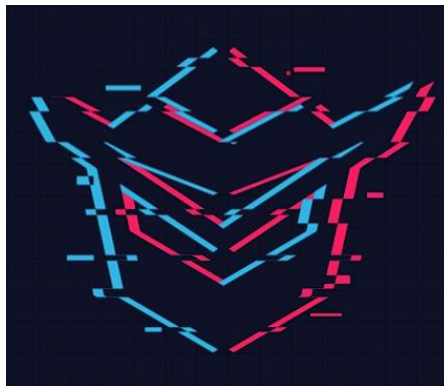


RogueGames
Object Design Document
Versione 0.3



Data: 07/02/2025

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/2025

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Lavorato Giovanni	0512116371
Battaglia Marco	0512116374
di Caprio Domenico	0512118543
De Vita Antonio	0512116944

Scritto da:	Giovanni Lavorato Marco Battaglia Domenico di Caprio Antonio De Vita
--------------------	---

Revision History

Data	Versione	Descrizione	Autore
16/12/2024	0.1	Stesura del Object Design Document (ODD)	Giovanni Lavorato Marco Battaglia Domenico di Caprio Antonio De Vita
10/01/2025	0.2	Aggiunta dell'introduzione e del package del sistema	Marco Battaglia
07/02/2025	0.3	Modifica dell'interfacce delle classi	Marco Battaglia Domenico di Caprio

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

Indice

1. Introduzione	4
1.1 Trade off.....	4
1.2 Linee guida.....	4
1.3 Definizione, acronimi, abbreviazioni	4
1.4 Reference.....	4
2. Package	5
2.1 Package com.RogueGames	5
3. Interfacce delle classi	8

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

1. Introduzione

1.1 Trade Off

-Usabilità vs. Sicurezza:

Optiamo per rendere il sito più facile da usare per i clienti, semplificando tutti i processi più importanti. Questa semplificazione comporterà, tuttavia, una riduzione della sicurezza.

-Tempo di rilascio vs. funzionalità

Optiamo per rilasciare il sistema in tempi brevi, focalizzandoci sugli aspetti più importanti, così da avere un prodotto già funzionale. Questa soluzione permetterà di rilasciare rapidi aggiornamenti, consentendo di migliorare e aggiungere nuove funzionalità al sistema.

-Tempo di risposta vs memoria

Optiamo per avere un sistema che risponde velocemente alle richieste, migliorando così l'esperienza e la soddisfazione dell'utente. Questo, tuttavia, influirà sullo spazio di memoria richiesto, aumentando il carico di dati in memoria.

1.2 Linee Guida

- Nomi dei metodi seguono le regole della convenzione camelCase;
- La prima parola dei metodi è sempre un verbo;
- Gli errori verranno gestiti tramite delle eccezioni senza valori di ritorno;

1.3 Definition, acronyms and abbreviations

RAD: Requirements Analysis Documentation

SSD: System Design Document

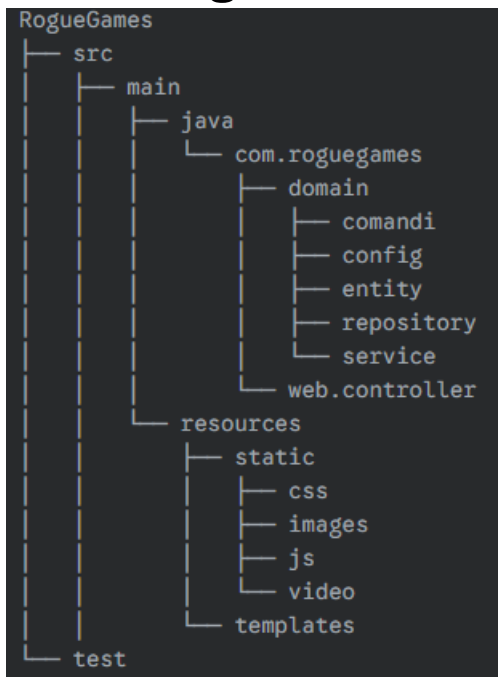
1.4 References

RAD_RogueGames

SDD_RogueGames

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

2. Package



All'interno del package RogueGames abbiamo:

-com.roguegames: contiene tutti i package che riguardano la logica di business e tutto il codice che gestisce l'interazione tra il sistema e l'utente;

-comandi; contiene i file che gestiscono l'implementazione del command design pattern;

1. AggiungiAlCarrello.java
2. ModificaQnt.java
3. RimuoviDalCarrello.java
4. AggiungiPreferitiCpmmand
5. RimuoviPreferitiCommand
6. Command.java
7. CommadManager

-config: contiene tutte le classi che servono per configurare il sistema;

1. CustomErrorController.java
2. GlobalExceptionHandler.java
3. WebConfig.java

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

-entity: contiene le classi che rappresentano le entità del dominio;

1. CartaDiCredito.java
2. IndirizzoSpedizione.java
3. IndirizzoSpedizioneId.java
4. Ordine.java
5. OrdineId.java
6. PCarrello.java
7. PCarrellId.java
8. Preferiti.java
9. PreferitiId.java
10. Prodotto.java
11. Tipologia.java
12. Utente.java

-repository: contiene le interfacce e le implementazioni che forniscono un'astrazione per l'accesso ai dati;

1. CarrelloRepository.java
2. CartaDiCreditoRepository.java
3. GestoreRepository.java
4. IndirizzoSpedizioneIdRepository.java
5. IndirizzoSpedizioneRepository.java
6. OrdineIdRepository.java
7. OrdineRepository.java
8. PCarrelloRepository.java
9. ProdottoRepository.java
10. UtenteRepository.java

-service: contiene la logica applicativa e coordina le operazioni tra le entità;

1. CarrelloService.java
2. CarrelloServiceImp.java
3. GestoreService.java
4. GestoreServiceImp.java
5. OrdineService.java
6. OrdineServiceImp.java
7. PreferitiService.java
8. PreferitiServiceImp.java
9. ProdottoService.java
10. ProdottoServiceImp.java
11. UtenteService.java
12. UtenteServiceImp.java

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

-controller: gestisce le richieste http in entrata e gestisce la restituzione delle risposte al client;

1. CarrelloController.java
2. CartaDiCreditoController.java
3. CatalogoController.java
4. DettagliProdottoController.java
5. GestoreAggiuntaController.java
6. GestoreController.java
7. HomeController.java
8. IndirizzoSpedizioneController.java
9. LoginController.java
10. OrdineController.java
11. PreferitiController.java
12. ProfiloController.java
13. ShowCarrelloController.java
14. ShowOrdiniController.java
15. UtenteController.java

- resources: contiene i file che non sono codice java compilato, ma che sono necessari all'applicazione;

-static: contiene tutte le risorse statiche;

- css
- images
- js
- video

-templates: contiene le pagine html

1. Carrello.html
2. Carte.html
3. Catalogo.html
4. DettagliProdotto.html
5. Errore.html
6. Footer.html
7. Header.html
8. Home.html
9. Indirizzi.html
10. Login.html
11. Logout.html
12. OrdineUtente.html

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

- 13. Preferiti.html
- 14. Prodotti.html
- 15. Profilo.html
- 16. Register.html
- 17. RiepilogoOrdine.html
- 18. StoricoOrdineGestore.html

- test: contiene tutte le classi necessarie per testare le varie componenti del sistema

- 19. CarrelloServiceIntegrationTest.java
- 20. CarrelloServiceTest.java
- 21. GestoreServiceIntegrationTest.java
- 22. GestoreServiceTest.java
- 23. OrdineServiceIntegrationTest.java
- 24. OrdineServiceTest.java
- 25. PreferitiServiceIntegrationTest.java
- 26. PreferitiServiceTest.java
- 27. RogueGamesTest.java
- 28. UtenteServiceIntegrationTest.java
- 29. UtenteServiceTest.java

3. Interfacce delle classi

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Registro Utente	verificaCredenziali(Email: String, Password: String)	pre: Email <> null and Pmail.matches("^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})\$") and and Password <> null Password.size() >= 9 and Password.matches(".*\\d.*") and Password.matches(".*[\\W_].*")	post: self.session <> null if login is successful
Registro Utente	registrati(Email: String, Password: String, Nome: String, Cognome: String, Età: Int, Indirizzo: String, Tel: String)	: Email <> null and Pmail.matches("^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})\$") and and Password <> null	post: self.username = Email and self.nome = Nome and

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

		Password.size() >= 9 and Password.matches(".*\\d.*") and Password.matches(".*[\\W_].*") and Nome <> null and Nome.size() >= 2 and Nome.size() <= 45 and Nome.matches("[a-zA-Z0-9]{2,45}\$") and Nome <> null and Cognome.size() >= 2 and Cognome.size() <= 45 and Cognome.matches("[a-zA-Z0-9]{2,45}\$") and Tel <> null and and Tel.matches("[0-9]{10}\$") and Eta <> null and Et >= 10 and Eta <= 120 <> null and Indirizzo <> null and and Indirizzo.size() >= 7 and Indirizzo.size() <= 45 and Indirizzo.matches("[a-zA-Z0-9]{7,45}\$")	self.cognome = Cognome
Registro Utente	modificaProfilo(Tel: String)	pre: Tel <> null and Tel.matches("[0-9]{10}\$")	post: self.telefono = Tel
Registro Utente	logOut()	pre: self.session <> null	

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Gestore	aggiuntaProdotto(Nome: String, Immagine: String, Video: String, Descrizione: String, CoV: Int, Prezzo: Double, CasaProd: String, Piattaforma: String,	pre: Nome <> null and Nome.size() > 0 and Nome.size() <= 45 and Prezzo <> null and and Prezzo > 0.01 and Immagine <>	post: self.prodotti includes new Prodotto

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

	<p>Genere: String, Tipo: String, DataRilascio: Date, Quantita: Int)</p>	<p>null and Descrizione <> null and CasaProd <> null and CasaProd.size() > 0 and CasaProd.size() <= 45 and Piattaforma <> null and (Piattaforma = "PlayStation" or Piattaforma = "Xbox" or Piattaforma = "Pc" or Piattaforma = "NintendoSwitch" or Piattaforma = "Nessuna" or Piattaforma = "") and Genere <> null and Genere.size() > 0 and Genere.size() <= 45 Tipo <> null and Tipo = "Videogiochi" or Tipo = "Console" or Tipo = "AF" or Tipo = "Accessori" and Quantita > 0 and DataRilascio <> null</p>	
Gestore	<p>modificaProdotto(Nome: String, Immagine: String,</p>	<p>pre: Nome <> null and</p>	<p>post: self.prodotti includes modified</p>

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

	Video: String, Descrizione: String, CoV: Int, Prezzo: Double, CasaProd: String, Piattaforma: String, Genere: String, Tipo: String, DataRilascio: Date, Quantita: Int)	Nome ∈ self.prodotti.Nome and Prezzo > 0.01 and Immagine <> null and Descrizione <> null and CasaProd <> null and Piattaforma <> null and Genere <> null and Tipo <> null and Quantita > 0 and DataRilascio <> null	Prodotto
Gestore	rimuoviProdotto(Nome: String)	pre: Nome <> null and Nome ∈ self.prodotti.Nome	post: self.prodotti excludes Prodotto

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Utente	aggiungiCartaCredito(cif: String, Scadenza: String, CVV: String, EmailUT: String)	pre: cif <> null and cif.size() >= 16 and cif.size() <= 18 and Scadenza <> null and and Scadenza.match es("^([0-9] 1[0- 2])\\d{2}\$") and CVV <> null and and	post: self.cartaCredito <> null

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

		CVV.matches("^\\d{3,4}\$") and EmailUT <> null	
Utente	aggiungiIndirizzoSpedizione(Provincia: String, CAP: Int, Via: String, Civico: String, Email: String, Citta: String)	pre: Provincia <> null and Provincia.matches("^([A-Z]{2})\$") and CAP > 0 and CAP.matches("^\\d{5}\$") and Via <> null and Via.size() >= 3 and Via.size() <= 50 and Via.matches("^([a-zA-Z0-9]{3,50})\$") and Civico <> null and Civico.size() >= 1 and Civico.size() <= 10 and Civico.matches("^([a-zA-Z0-9]{1,10})\$") Email <> null and Email.matches("^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.([a-zA-Z]{2,6})\$") Citta <> null	post: self.indirizzoSpedizione <> null
Utente	rimuovereCartaCredito(cif: String)	pre: cif <> null	post: self.cartaCredito = null
Utente	rimuovereIndirizzoSpedizione (Provincia: String, CAP: Int,	pre: Provincia <> null and CAP > 0	post: self.indirizzoSped

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

	Via: String, Civico: String, Email: String, Citta: String)	and Via <> null and Civico <> null and Email <> null and Citta <> null and IndirizzoSpedizione->exists	izione = null
Utente	ModificaIndirizzoSpedizione(Provincia: String, CAP: Int, Via: String, Civico: String, Email: String, Citta: String)	pre: Provincia <> null and Provincia.matches("^([A-Z]{2})\$") and CAP > 0 and CAP.matches("^\\d{5}\$") and Via <> null and Via.size() >= 3 and Via.size() <= 50 and Via.matches("^([a-zA-Z0-9]{3,50})\$") and Civico <> null and Civico.size() >= 1 and Civico.size() <= 10 and Civico.matches("^([a-zA-Z0-9]{1,10})\$") Email <> null and Email.matches("^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.([a-zA-Z]{2,6})\$") Citta <> null	post: self.indirizzoSpedizione <> null
Utente	ModificaCartaCredito((cif:	pre: cif <> null	post:

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

	String, Scadenza: String, CVV: String, EmailUT: String)	and cif.size() >= 16 and cif.size() <= 18 and Scadenza <> null and Scadenza.matches("^(0[1-9] 1[0-2])\\d{2}\$") and CVV <> null and CVV.matches("^\\d{3,4}\$") and EmailUT <> null	self.cartaCredito <> null
--	---	--	---------------------------

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Prodotto	ordinamentoPerData(DataRilascio: String)	pre: DataRilascio <> null	post: self.prodotti = self.prodotti->sortedBy(p)
Prodotto	filtraggioPiattaforma(Piattaforma: String)	pre: Piattaforma <> null	post: self.prodotti = self.prodotti->select(p)

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Carrello	aggiungiACarrello(Nome: String, Email: String)	pre: Nome <> null and Nome ∈ self.prodotti.Nome and Email <> null and Email ∈ utente.Email	post: self.carrello->exists(c)
Carrello	modificaQntCarrello(NomeProdotto:	pre: NomeProdotto <> null	post:

Progetto: RogueGames	Versione: 0.3
Documento: Object Design Document	Data: 07/02/25

	String, quantitaStr: String)	quantitaStr <> null quantitaStr > 0 and quantitaStr.matches("[1-9]+") and Integer.parseInt(quantitaStr) > 0	self.carrello->exists(c)
Carrello	RimuoviAlCarrello (NomeProdotto: String)	pre: NomeProdotto <> null	post: self.carrello = self.carrello->select(c)

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Preferiti	rimuoviProdottoPreferiti(NomeProdotto: String)	pre: NomeProdotto <> null	post: self.preferiti = self.preferiti->select(p)
Preferiti	aggiuntaPreferiti(Nome: String, Email: String)	pre: Nome <> null and Email <> null	post: self.preferiti->exists(p)

Classe	Metodo	Pre-condizioni (OCL)	Post-condizioni (OCL)
Ordine	processaOrdine(Carrello: list[Carrello],tot: Double,EmailUT:String)	Pre: Carrello <> null and CartaCredito <> null and IndirizzoSpedizione<>null and Utente.ruolo <> "gestore"	Post: o.exists
Ordine	getOrdine()	pre: self.session <> null and Utente.ruolo.matches("[gestore]")	
Ordine	getOrdineUtente(Email: String)	pre: self.session <> null and Utente.ruolo <> "gestore"	