



**Università degli
Studi di Salerno**



iSnapGaming

Software Engineering Project

Team



Marco Santoriello



Chiara Coscarelli

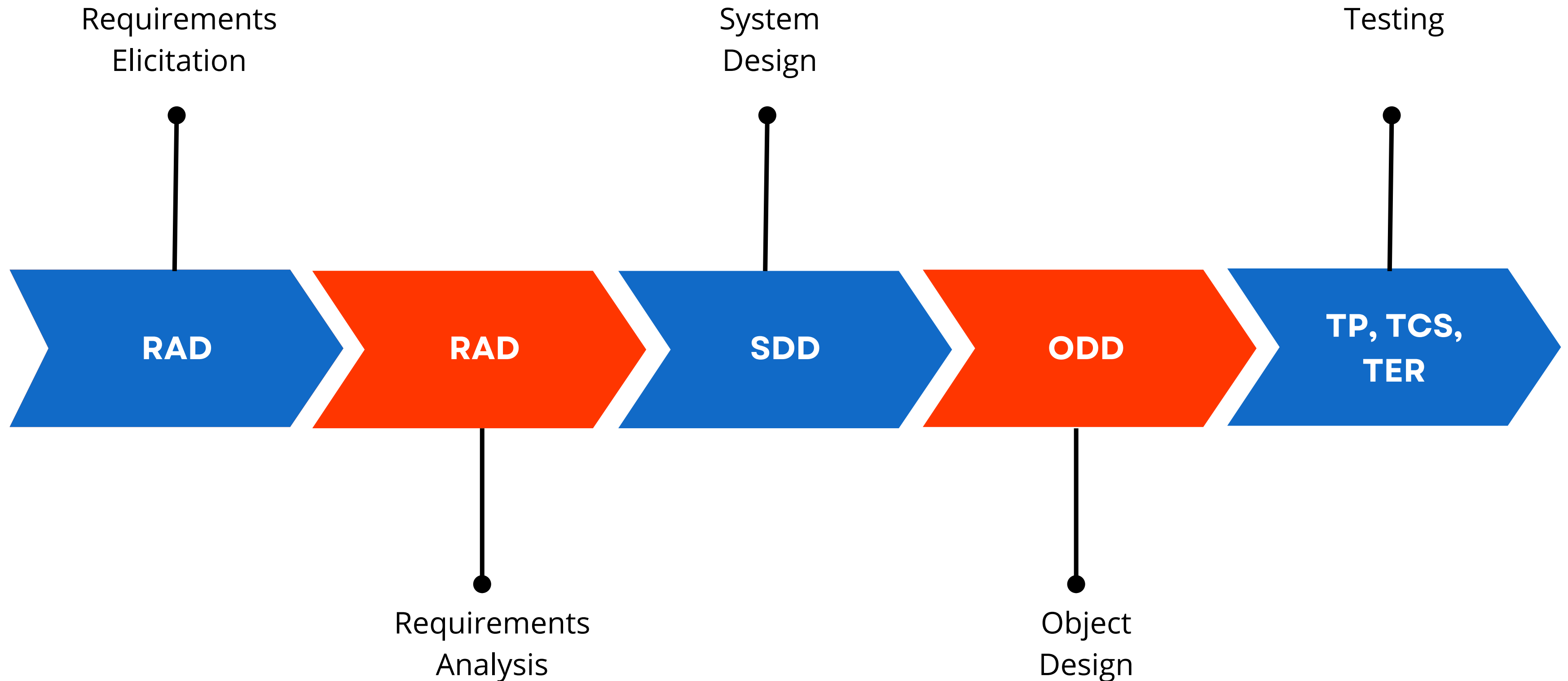



Fabio Sessa

Introduzione



Fasi dello Sviluppo





ANALISI DEI REQUISITI

Attori

Managers



Product
Manager



Order
Manager



Review
Manager



Executive
Coordinator

Users



Ospite



Cliente

Esterni



Servizio di
autorizzazione
pagamento

Entità

Manager

User

Customer

Address

Order

Product

Cart

Review

Entità

Manager

User

Customer

Address

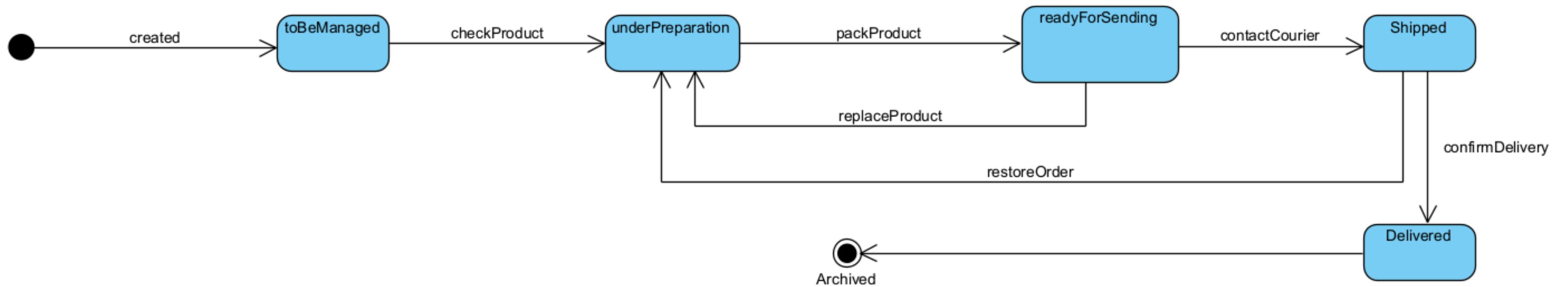
Order

Product

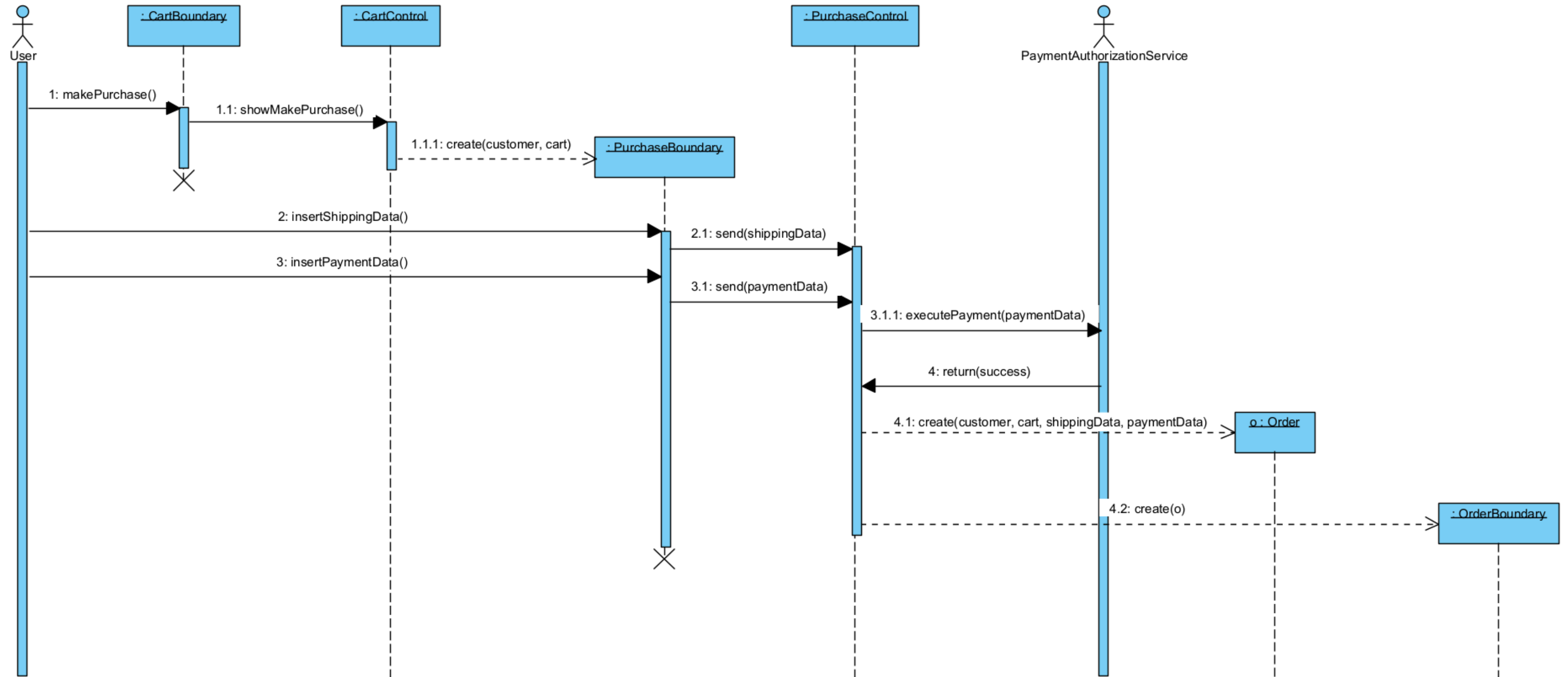
Cart

Review

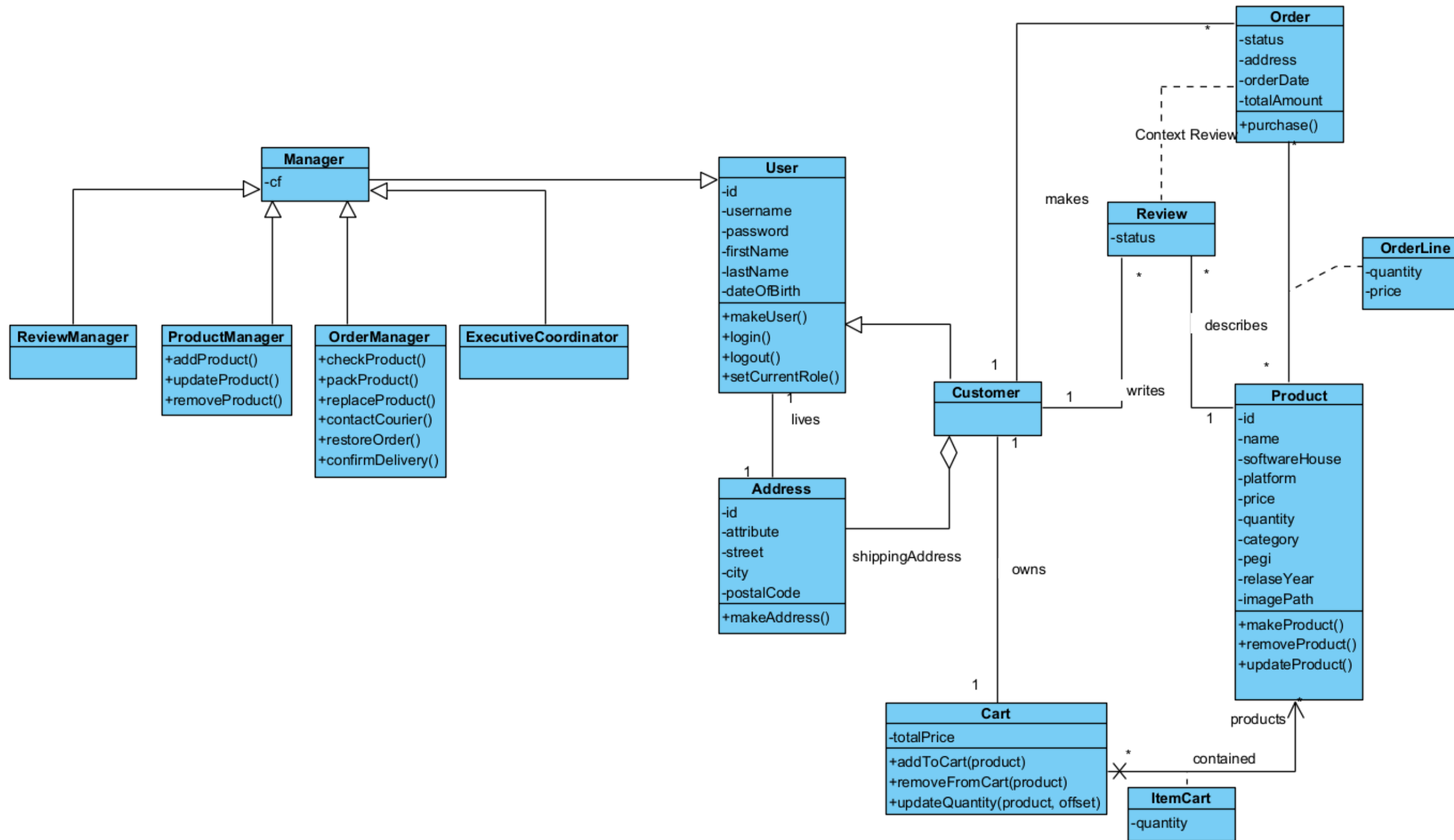
Ordine



Effettuazione Acquisto



Object Model





SYSTEM DESIGN

Architettura



PRESENTATION

Architettura



PRESENTATION

APPLICATION LOGIC

Architettura

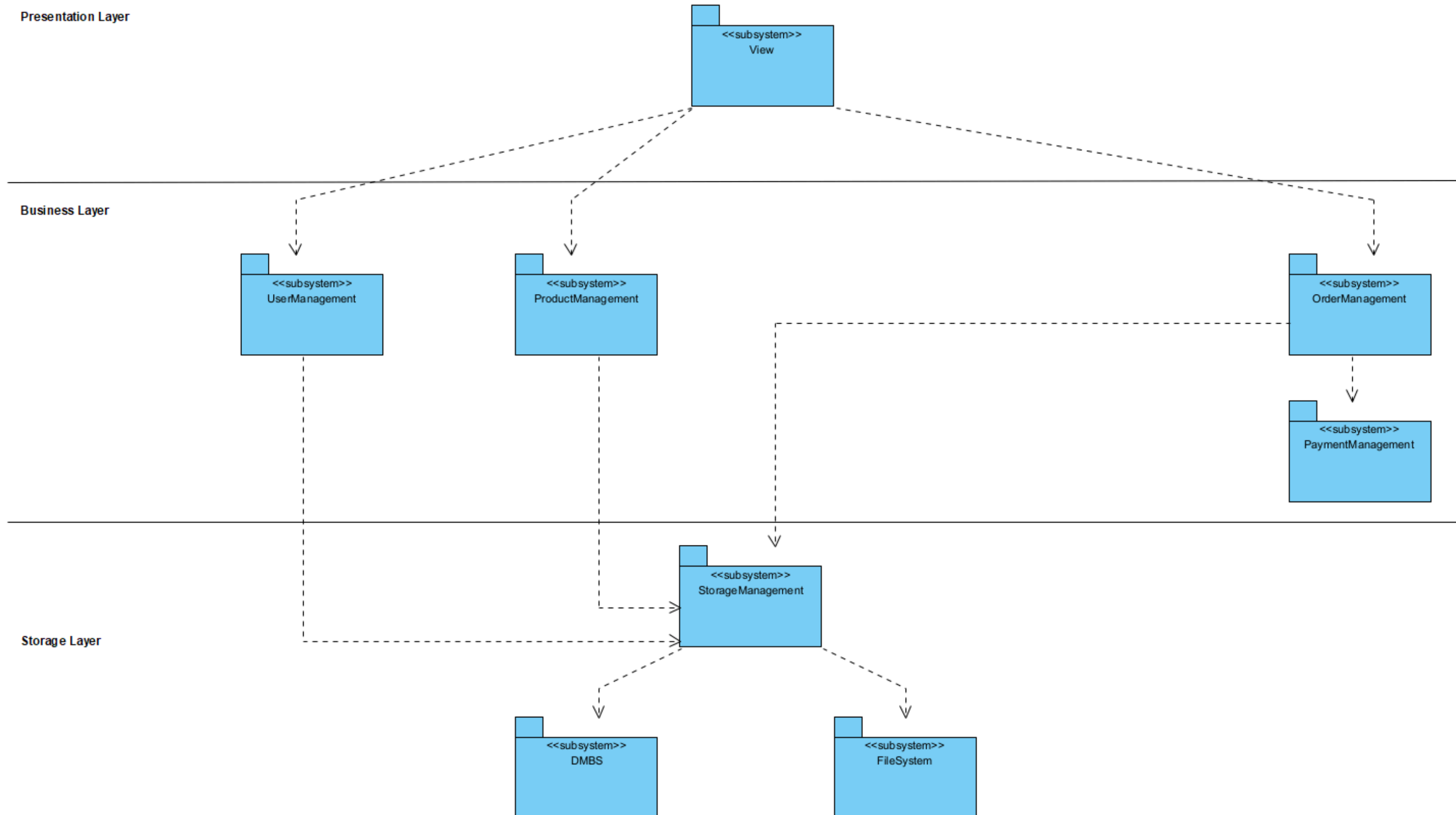


PRESENTATION

APPLICATION LOGIC

STORAGE LAYER

Decomposizione del sistema



Decomposizione del sistema

View

Interazione Utente
Pagine Web
Servlet

User Management

Registrazione Utente
Gestione Utente
Autenticazione Utente

Product Management

Gestione Prodotto

Order Management

Creazione Ordine
Gestione Ordine
Carrello

Payment Management

Pagamento
Comunicazione
Sistema Esterno

Storage Management

Interazione con DBMS
e File System
Gestione Persistenza

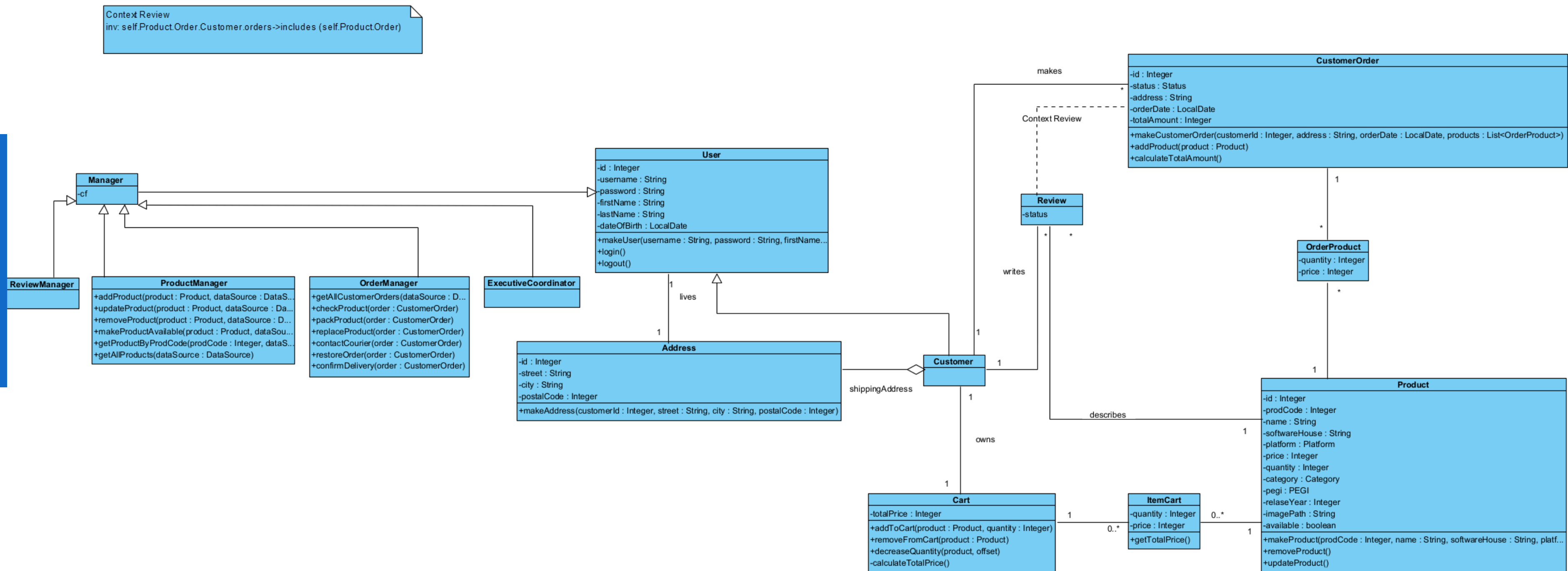


OBJECT DESIGN

Specifica Interfacce

ItemCart	
Description	Classe che rappresenta un particolare prodotto nel carrello di un customer.
Invariant	context ItemCart inv: cart<> null & product <> null & quantity > 0
getTotalPrice	Metodo che calcola il costo totale di un prodotto in relazione alla quantità inserita. context ItemCart::getTotalPrice() pre: - context ItemCart::getTotalPrice() post: getTotalPrice() = this.price * this.quantity

Ottimizzazione Object Design Model



Effettuazione Pagamento

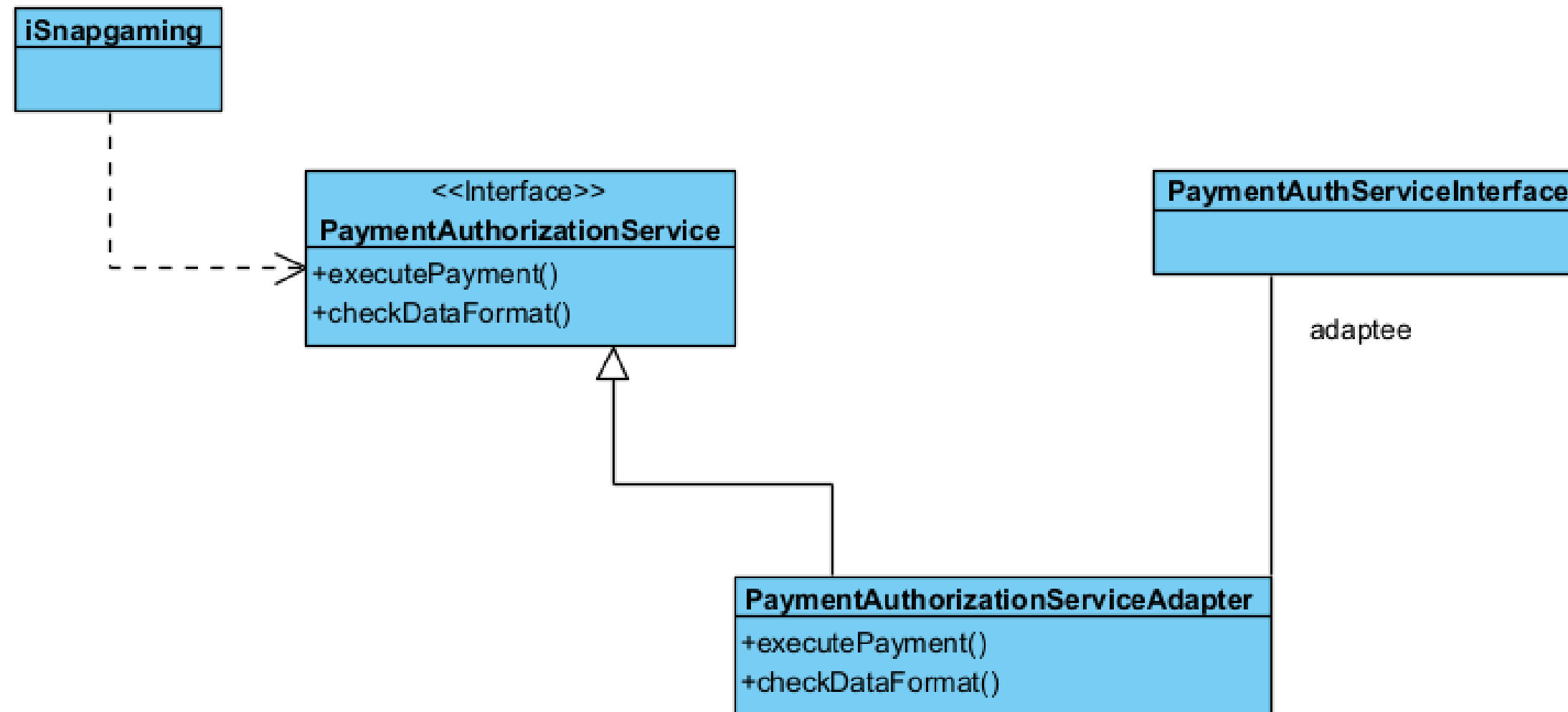
Ci si affida ad un sistema esterno per la finalizzazione del pagamento

Tuttavia, vogliamo rendere il nostro meccanismo indipendente dall'interfaccia del servizio esterno

Come?



Soluzione: Adapter Pattern





TESTING

Testing

Test Plane



Test Execution
Report



Test Case
Specification



Testing



Testing Unità



Testing di integrazione

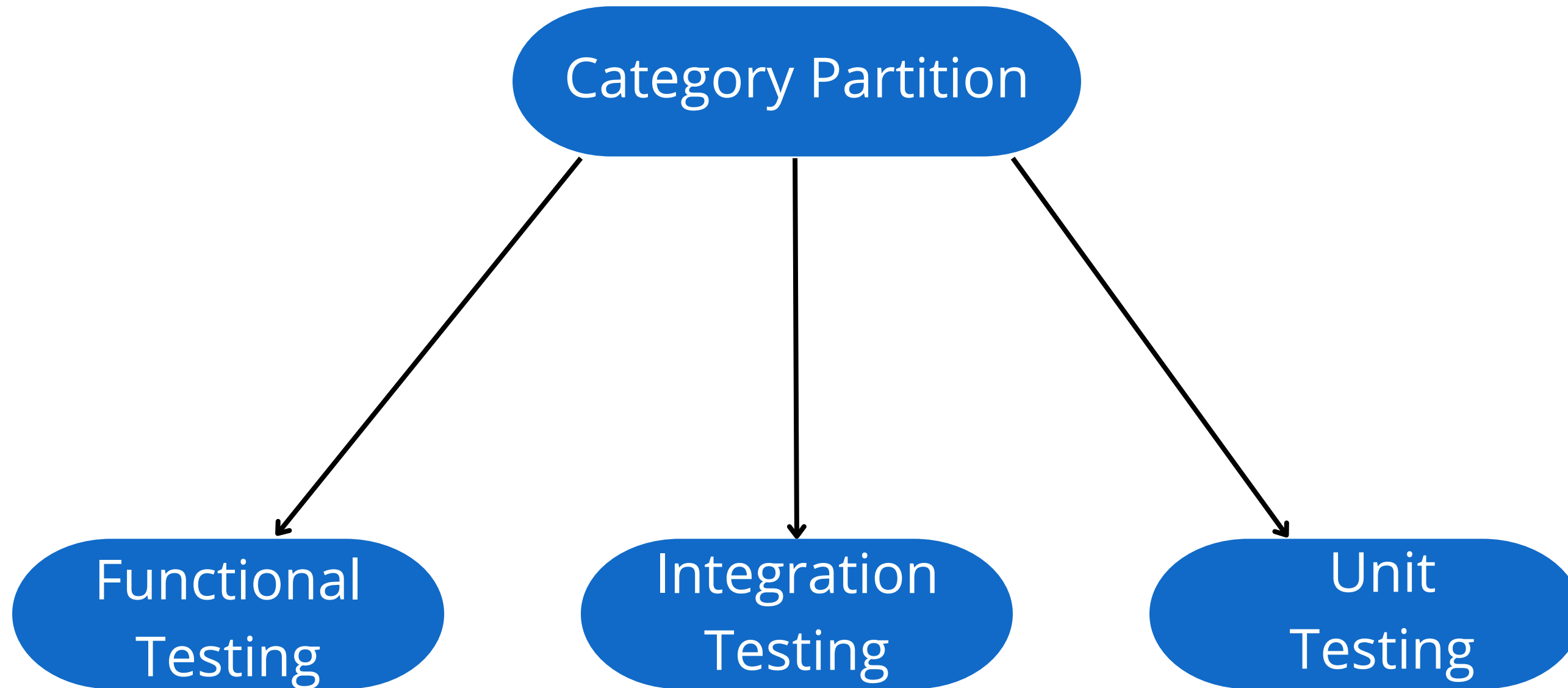


Testing di sistema

Tecniche utilizzate

- I test di Unità sono stati effettuati sui metodi delle classi sviluppate mediante JUnit e Mockito
- I test di Integrazione sono stati effettuati sulle classe che contengono dipendenza con altre classi
- I test di Sistema è stato realizzato tramite il tool Selenium

Metodologia



Registrazione

Registrazione

PARAMETERS	firstName, lastName, dateOfBirth, email, password, conferma password, street, city, postalCode
OBJECTS OF THE ENVIRONMENT	Database
CATEGORIES	CHOICES
Formato firstName (sole lettere)	F1: firstName rispetta il formato del nome [property firstNameVal] F2: firstName non rispetta il formato del nome
Formato lastName (sole lettere)	L1: lastName rispetta il formato del cognome [property lastNameVal] L2: lastName non rispetta il formato del cognome [if firstNameVal]
Valore dateOfBirth (non può essere nel futuro)	D1: dateOfBirth corretta [property dateOfBirthVal] D2: dateOfBirth non corretta [if firstNameVal & lastNameVal]
Formato email per username	E1: email corretta [property emailVal] E2: email non corretta [if firstNameVal & lastNameVal & dateOfBirthVal]
Lunghezza password (>= 8)	LP1: lunghezza password valida [property lenPwVal] LP2: lunghezza password non valida [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal]
Formato password (maiuscole, minuscole, cifre)	P1: password rispetta il formato [property validPwFormat] P2: password non rispetta il formato [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPwVal]
Valore confirmPassword	CP1: Conferma password corrisponde a password [property pwMatch] CP2: Conferma password non corrisponde a password [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat]
Formato City (solo caratteri)	C1: city rispetta il formato city [property cityVal] C2: city non rispetta il formato city [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat & pwMatch]
Formato postalCode (len = 5, solo cifre)	PC1: postalCode rispetta il formato postalCode [property postalCodeVal] PC2: postalCode non rispetta il formato postalCode [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat & pwMatch]
Username non presente nel database	U1: Username non presente nel database U2: Username presente nel database [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat & pwMatch & postalCodeVal]



**Università degli
Studi di Salerno**



**GRAZIE PER
L'ATTENZIONE!**