

Università degli studi di Napoli Parthenope

Progetto Ingegneria del Software e Interazione

Uomo Macchina

Documento di progettazione

Track Me



D'ANGELO SIMONE 0124/002601
VETRANO ALESSIO 0124/002326
GALLUCCIO LUIGI 0124/002293

Prof.ssa Barra
Prof.ssa Staffa

Sommario

1. Introduzione	1
1.1 Scopo del sistema	2
1.2 Obiettivi di progettazione	2
1.3 Riferimenti	2
2. Sistema	3
2.1 Decomposizione del sistema	3
2.2 Hardware/Software mapping	4
2.3 Gestione dei dati persistenti	4
2.4 Controllo accessi e sicurezza	5
2.5 Condizioni limite	5

1. Introduzione

1.1 Scopo del sistema

Lo scopo del sistema è quello di gestire e ottimizzare il carico e il rintracciamento delle merci per un'azienda di trasporto che consegna in diverse sedi in Italia. Il sistema utilizza un algoritmo genetico per scegliere il veicolo più adatto per ogni collo, tenendo conto della sua capienza e del peso. Inoltre, il sistema mantiene traccia dello stato del collo durante la consegna e permette ai destinatari di rintracciare la merce tramite il codice del collo.

Il software include molteplici funzionalità:

- Inviare un pacco (con specifica sul peso) inserendo i vari mittenti e destinatari
- Localizzare un pacco mediante proprio codice identificativo
- Modificare la sede associata al pacco o impostare quest'ultimo come stato consegnato
- Visualizzare lo storico dei pacchi inviati durante l'utilizzo del nostro software

1.2 Obiettivi di progettazione

Nel nostro progetto abbiamo deciso di utilizzare JavaFX come strumento principale per la realizzazione delle interfacce grafiche.

Per aiutare nella realizzazione della interfaccia grafica, abbiamo deciso di utilizzare un tool compatibile con JavaFX chiamato "Scene Builder". Questo è uno strumento di layout visivo che consente agli utenti di progettare rapidamente interfacce utente senza dover scrivere codice. Ciò ha ulteriormente semplificato la realizzazione dell'applicazione, rendendola più intuitiva e facile da utilizzare.

1.3 Riferimenti

Per quanto concerne la progettazione dell'applicazione abbiamo utilizzato l'architettura MVC (Model-View-Controller) ovvero quel pattern utilizzato per lo sviluppo di interfacce utente separando la logica dell'applicazione:

- **Model:** rappresentazione dei dati e logica di business.
- **View:** Rappresentazione dell'interfaccia utente.
- **Controller:** Gestione delle interazioni dell'utente con l'applicazione.

Inoltre, l'applicazione trova una sua risoluzione nell'implementazione di un particolare **algoritmo**, ossia quello **genetico**. Quest'ultimo è un metodo di ottimizzazione basato su principi ispirati alla biologia evolutiva, come la selezione naturale e la riproduzione con mutazione. Viene utilizzato in una vasta gamma di campi, tra cui l'intelligenza artificiale, o la robotica. In generale, questo tipo di algoritmo utilizza una popolazione di soluzioni candidate, che evolvono attraverso diverse generazioni attraverso processi di selezione, crossover e mutazione.

Inoltre abbiamo adattato i vari principi **S.O.L.I.D** che rappresentano un insieme di regole che aiutano a creare software che sia facile da mantenere, estendere e testare. Aderendo a questi principi, un progetto può avere:

- Codice più leggibile e comprensibile
- Maggiore flessibilità per future modifiche
- Minor numero di bug
- Migliore qualità del codice

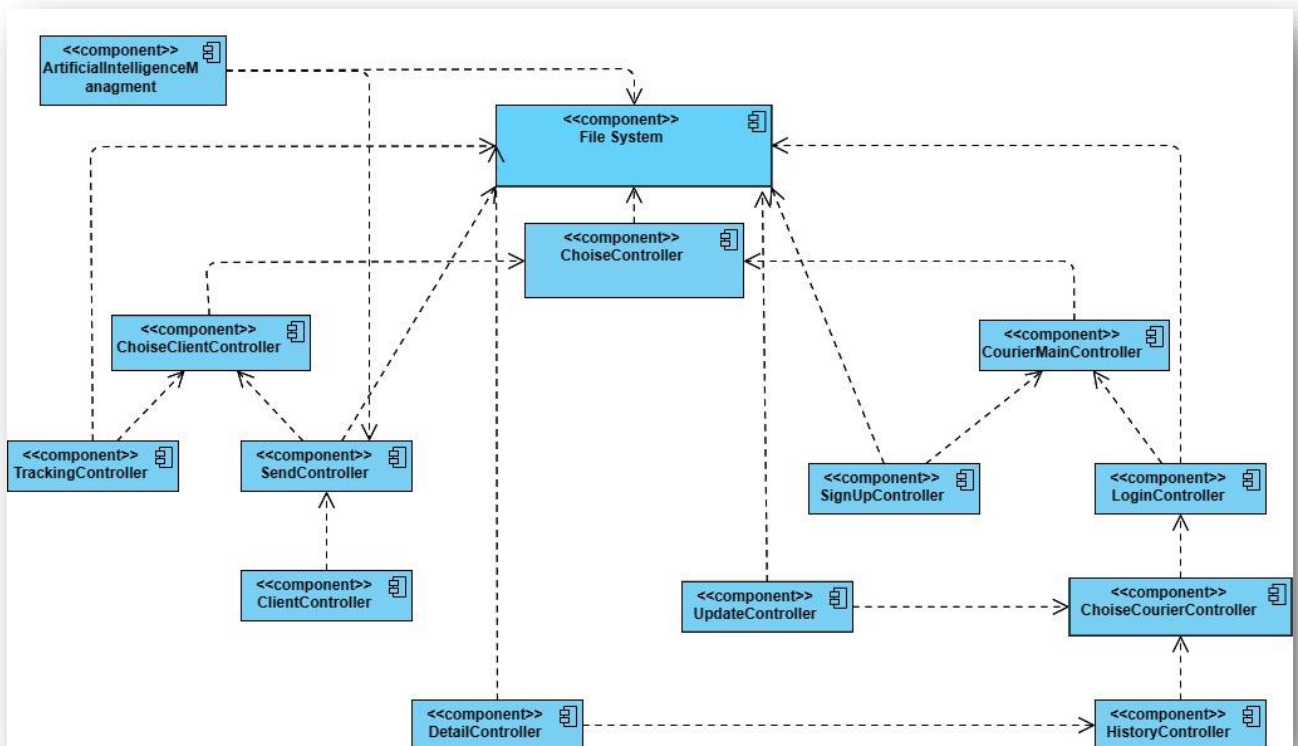
- Maggiore capacità di collaborazione tra sviluppatori.

Inoltre, per la progettazione abbiamo seguito diversi design pattern, quelli utilizzati in fase di progettazione sono:

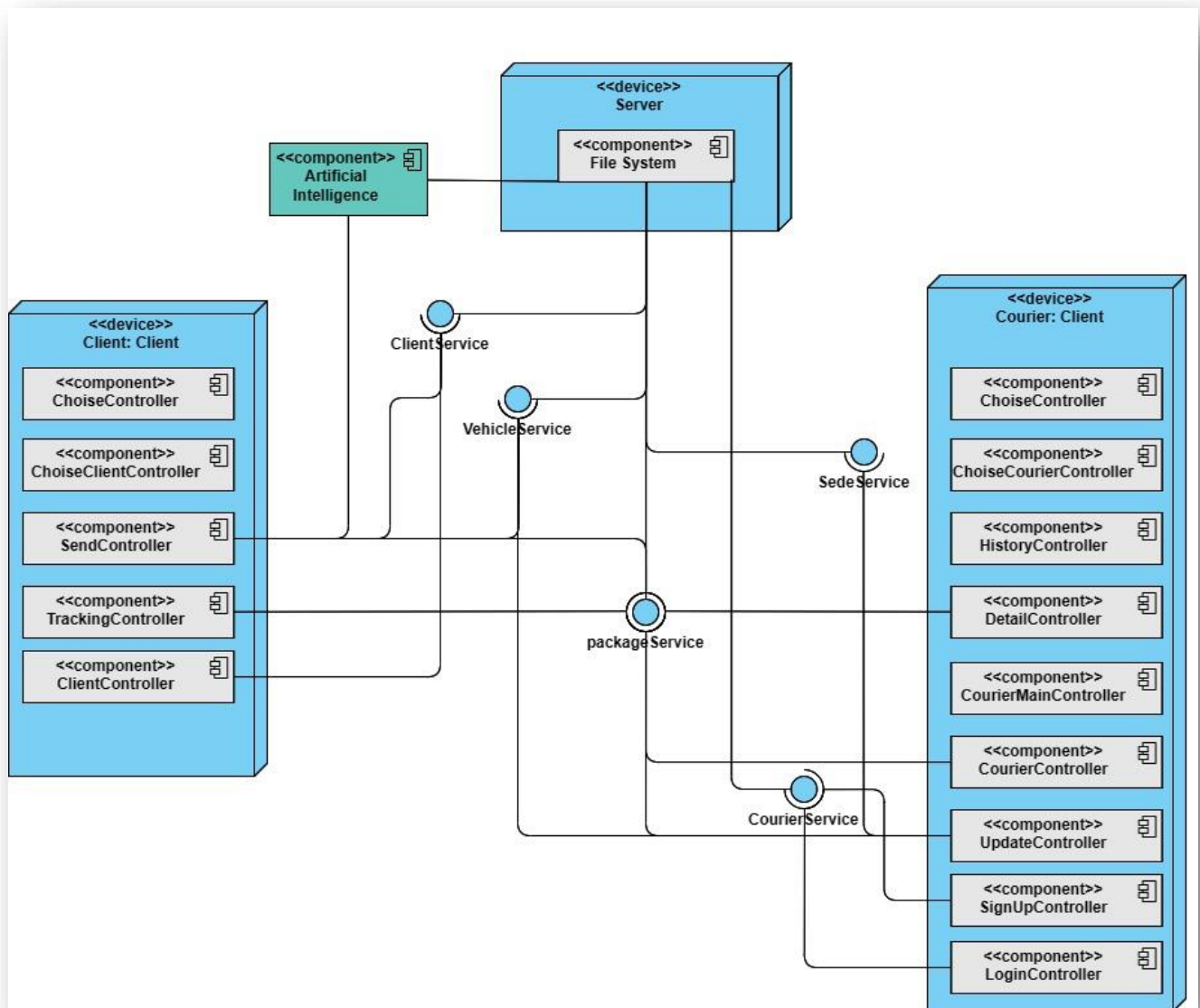
- Template
- Adapter
- Factory Pattern
- Prototype
- Builder
- Flyweight

2. Sistema

2.1 Decomposizione del sistema



2.2 Hardware/Software mapping



2.3 Gestione dei dati persistenti

I dati persistenti del progetto sono memorizzati all'interno di file *.csv (Comma Separated Values).

L'utilizzo di questa tipologia di file è dovuto a diverse ragioni:

- **Semplicità:** I file CSV sono semplici da gestire e possono essere facilmente aperti e modificati con un editor di testo o un foglio di calcolo. Non è necessario un software di gestione del database per utilizzare i file CSV.
- **Portabilità:** I file CSV possono essere facilmente condivisi e utilizzati su diverse piattaforme e sistemi operativi senza la necessità di configurazioni o driver specifici.
- **Accessibilità:** I file CSV possono essere facilmente aperti e utilizzati da molteplici applicazioni e linguaggi di programmazione, tra cui Java.

- **Costo:** Utilizzare un file CSV potrebbe essere meno costoso rispetto all'utilizzo di un database relazionale, soprattutto per piccole e medie quantità di dati.

2.4 Controllo accessi e sicurezza

Oggetti/Attori	Corriere	FileSystem	Cliente
Sign-up Courier	onSubmitSignup()	onSubmitSignUp()	
Log-in Courier	onSubmitLogin()	onSubmitLogin()	
UpdateStatusPack	updateStatus()	update()	
TrackPack		selectRecord()	onTrace()
HistoryView	getVehicle() getWeight() getSite() getID()	selectRecord()	
SendPack		update() selectRecord()	send() sendVehicles()

2.5 Condizioni limite

- **Prima accensione del sistema:** le sedi e i veicoli utilizzati all'interno dell'applicazione vengono popolati con i dati dell'azienda di logistica alla prima accensione del sistema.
- **Registrazione del Cliente:** Il cliente può gestire l'inserimento dei dati del mittente e del destinatario per mandare un collo.