



SDD

System Design Document

SmartCargo

Riferimento	C07_SOW_Educational_SmartCargo_V2.0 , C07_RAD_SmartCargo_V2.0
Versione	2.2
Data	12/12/2023
Destinatario	Prof.ssa Filomena Ferrucci (FF)
Presentato da	Mariapia Sorrentino (M.S.) - 0512113750 Roksana Duda (R.D.) – 0512114326 Francesco F. Ambrosio (F.A.) - 0512114152 Paolo Murino (P.M.) - 0512116057 Amedeo Napolitano (A.N.) - 0512111956 Andreea C. C. Oprisescu (A.O.) - 0512114104
Approvato da	Vincenzo Esposito (VE) Nicola Tortora (NT)



Sommario

REVISION HISTORY	3
1. INTRODUZIONE	4
1.1 SCOPO DEL SISTEMA	4
1.2 DESIGN GOALS	4
1.3 DEFINIZIONE, ACRONIMI E ABBREVIAZIONI	5
1.4 RIFERIMENTI	6
1.5 PANORAMICA	6
2. ARCHITETTURA SOFTWARE CORRENTE	7
3. SOFTWARE PROPOSTO	8
3.1. PANORAMICA	8
3.2. DECOMPOSIZIONE IN SOTTOSISTEMI	9
SOTTOSISTEMA INGRESSO	12
SOTTOSISTEMA MONITORAGGIO	12
SOTTOSISTEMA OPERAZIONE CARICO/SCARICO	13
SOTTOSISTEMA AUTENTICAZIONE	13
SOTTOSISTEMA ISSUE	14
SOTTOSISTEMA USCITA	14
SOTTOSISTEMA ACCOUNT	15
SOTTOSISTEMA COMUNICAZIONE	15
3.3. MAPPING HARDWARE/SOFTWARE	16
3.4. GESTIONE DEI DATI PERSISTENTI	17
ENTITY CLASS DIAGRAM – AUTORI: MARIAPIA SORRENTINO, PAOLO MURINO, FRANCESCO F. AMBROSIO	18
SCHEMA ER RISTRUTTURATO – AUTORI: TUTTI I T.M.	19
DIZIONARIO DEI DATI	20
3.5. CONTROLLO DEGLI ACCESSI DI SICUREZZA	27
3.6. CONTROLLO GLOBALE DEL SOFTWARE	29
3.7. CONDIZIONI LIMITE	30
4. SERVIZI DEI SOTTOSISTEMI	34
SOTTOSISTEMA AUTENTICAZIONE	34
SOTTOSISTEMA INGRESSO	34
SOTTOSISTEMA MONITORAGGIO	34
SOTTOSISTEMA OPERAZIONE CARICO/SCARICO	35
SOTTOSISTEMA ISSUE	35
SOTTOSISTEMA COMUNICAZIONE	36
SOTTOSISTEMA USCITA	36
SOTTOSISTEMA ACCOUNT	36
5. GLOSSARIO	37



Revision History

Data	Versione	Descrizione	Autori
30/10/2023	1.0	Definizione del Template.	V.E.
30/11/2023	1.1	Aggiunta della prima sezione e dei design goals.	M.S., P.M., A.O., R.D., F.A., A.N.
30/11/2023	1.2	Aggiunta sezione 2.	F.A.
01/12/2023	1.3	Aggiunta sezione 3 fino alla decomposizione in sottosistemi compresa.	M.S., P.M., A.O., R.D., F.A., A.N.
01/12/2023	1.4	Aggiunta Mapping H/S.	F.A., A.N.
02/12/2023	1.5	Aggiunta gestione dei dati persistenti.	M.S., P.M., A.O., R.D., F.A., A.N.
03/12/2023	1.6	Aggiunta controllo degli accessi di sicurezza e condizioni limite.	P.M., M.S., A.O.
05/12/2023	1.7	Aggiunta controllo globale del SW.	R.D., A.N., F.A.
09/12/2023	1.8	Aggiunta sezione 5	M.S., A.O.
13/10/2023	2.0	Revisione per consegna intermedia	Tutti
20/01/2024	2.1	Revisione architettura del sistema	AN. , FA.
22/01/2024	2.2	Revisione per consegna finale	P.M, R.D.,M.S.,A.O.



1. Introduzione

1.1 Scopo del sistema

Il sistema proposto mira a ottimizzare e semplificare le operazioni di gestione dei trasporti all'interno del porto. Il suo scopo fondamentale è consentire un controllo efficiente e completo attraverso strumenti di identificazione, monitoraggio, registrazione e gestione delle attività. Inoltre, saranno inclusi strumenti di comunicazione per gestire le interazioni tra operatori e autotrasportatori, facilitando la segnalazione e consentendo anche di gestire in modo efficiente le problematiche rilevate. Il sistema impiegherà una tecnologia di tracciamento per definire e confrontare costantemente la posizione reale dell'autotrasportatore con un percorso ottimale. In caso di deviazioni significative, il sistema genererà immediatamente un avviso.

Questo approccio non solo garantisce il monitoraggio sicuro del trasporto, ma consente anche interventi rapidi in situazioni critiche.

1.2 Design Goals

I Design Goals possono essere suddivisi in diverse categorie:

- **Dependability:** racchiude tutti i requisiti che riguardano la minimizzazione dei crash e le conseguenze che essi provocano sul sistema.
- **Maintenance:** racchiude tutti i requisiti che riguardano la manutenibilità e la modifica del sistema dopo il suo rilascio.
- **Performance:** racchiude tutti i requisiti imposti sul sistema in termini di spazio e velocità.
- **End user:** racchiude tutti i requisiti desiderabili dal punto di vista dell'utente.

Ogni Design Goal è definito da vari attributi:

- **Rank,** definisce la priorità di ogni Design Goal, rappresentata da un valore che va da 1 a 18, dove 1 rappresenta la priorità massima, mentre 18 la priorità minima.
- **Nome,** definisce un identificatore univoco del Design Goal.
- **Descrizione,** definisce una breve spiegazione del Design Goal preso in considerazione.
- **Categoria,** definisce la categoria di appartenenza del Design Goal, selezionabile da uno dei sopra elencati.



- **Origine**, definisce quali RNF hanno generato il Design Goal.

Design goals:

Per consultare i design goals del sistema, seguire il seguente link: [C07 DesignGoals SmartCargo](#)

Trade-off

Trade-off	Descrizione
Massima compatibilità vs nuove funzionalità	Per garantire la massima compatibilità con il sistema già in uso si andranno ad individuare solo le funzionalità essenziali da apportare, che però consentano di agevolare e migliorare l'efficienza delle operazioni.
Memorie efficienti vs costi	Per garantire un'efficiente gestione dei dati durante le operazioni di carico/scarico e monitoraggio, daremo priorità all'utilizzo di memorie di ultima generazione, anche a discapito della minimizzazione dei costi.
Tempo di rilascio vs funzionalità	Se i tempi di rilascio sono stringenti, possono essere rilasciate meno funzionalità di quelle richieste, ma nei tempi giusti.

1.3 Definizione, Acronimi e Abbreviazioni

Di seguito sono riportate alcune definizioni presenti nel documento:

- **Design Goals:** le caratteristiche su cui il sistema deve concentrarsi.
- **Trade-off:** analisi delle decisioni di progettazione che implicano compromessi tra obiettivi concorrenti.
- **Sottosistema:** un insieme di servizi all'interno del dominio applicativo, collegati da una relazione funzionale.
- **Mapping Hardware/Software:** analisi della correlazione tra le componenti fisiche e logiche costituenti il sistema.
- **Dati Persistenti:** dati che sopravvivono anche dopo che il programma che li ha generati è stato eseguito e vengono, di conseguenza, memorizzati.
- **DBMS:** Database Management System.



- **RAD:** Requirements Analysis Document.

1.4 Riferimenti

Documenti di riferimento, utili alla comprensione del sistema proposto:

- **Statement of Work:** [C07 SOW Educational SmartCargo V2.0](#)
- **Requirements Analysis Document:** [C07 RAD SmartCargo V2.0](#)

1.5 Panoramica

Il documento è diviso in quattro sezioni:

- **Architettura Software Corrente:** Descrive il sistema corrente per consentire un confronto con il sistema proposto.
- **Architettura Software Proposta:** Descrive dettagliatamente il sistema software che si intende implementare.
- **Servizi dei sottosistemi:** Comprende tabelle che elencano i servizi forniti da ciascun sottoinsieme del sistema.
- **Glossario:** Raccoglie i termini specifici del settore applicativo utilizzati all'interno del SDD.



2. Architettura Software Corrente

Attualmente, il sistema informatico del Porto si basa su una struttura limitata, focalizzata principalmente su un database ridotto destinato a registrare le prenotazioni degli autotrasportatori per l'accesso al porto. Questo database memorizza informazioni fondamentali, come l'identità dell'autista, gli orari di arrivo e partenza, insieme alla tipologia e quantità della merce trasportata.

Tuttavia, l'architettura attuale manca delle funzionalità necessarie per raggiungere gli obiettivi delineati nel progetto proposto.

La mancanza di un sistema di monitoraggio e tracciamento dei percorsi dei veicoli all'interno del porto impedisce l'implementazione di molte caratteristiche chiave in termini di sicurezza ed efficienza.

Tra le principali lacune vi sono:

- Il supporto per la definizione di percorsi ottimizzati.
- Il tracciamento in tempo reale dei veicoli.
- La valutazione dei percorsi effettuati.
- La possibilità di ottenere informazioni sui veicoli transitati in un determinato periodo.

In breve, l'attuale architettura software risulta carente se confrontata con una nuova proposta strutturata. La nuova proposta si prefigge di colmare queste lacune, offrendo un'architettura più avanzata e funzionale, finalizzata a potenziare la sicurezza, l'efficienza e la sostenibilità complessiva delle operazioni portuali.



3. Software Proposto

L'obiettivo principale del progetto è l'implementazione di un sistema integrato volto a potenziare le infrastrutture portuali e migliorare la sicurezza ed efficienza delle operazioni di carico e scarico delle merci nel Porto di Valencia. In particolare, mira a rafforzare il controllo sulle attività portuali, prevenire il trasporto illegale di merci e ottimizzare i percorsi di transito al fine di ridurre le emissioni di CO2 e mitigare l'impatto ambientale generale. Deve supportare:

- la fase di definizione dei percorsi corretti ed efficienti all'interno del porto per ogni camion,
- la fase di tracciamento in tempo reale di un camion all'interno del porto,
- la fase di valutazione dei percorsi compiuti dai camion all'interno del porto,
- la ricerca di informazioni dei camion transitati in un determinato periodo,
- le fasi di valutazione e report delle attività,
- un sistema di alerting per gli operatori della sicurezza,
- un canale di comunicazione tra l'operatore della sicurezza e l'autista.

3.1. Panoramica

L'architettura proposta seguirà uno schema basato su servizi, adottando il modello di architettura a servizi per il nostro software. Tale scelta è stata motivata dalla necessità di rispondere in modo flessibile ed efficiente alle richieste del nostro progetto, soprattutto considerando il contesto delle web application. L'approccio a servizi facilita una chiara separazione delle responsabilità, migliorando vari aspetti critici del sistema. Tuttavia, è fondamentale considerare anche gli aspetti negativi di questa architettura, tra cui:

- Complessità iniziale: L'implementazione potrebbe richiedere uno sforzo iniziale maggiore, specialmente in progetti di dimensioni ridotte. La comprensione della struttura e dell'organizzazione dei vari servizi potrebbe richiedere tempo e risorse aggiuntive.
- Apprendimento iniziale: Se il team non fosse già esperto nell'utilizzo dell'architettura a servizi, potrebbe richiedere un periodo di apprendimento iniziale. Questo potrebbe ritardare l'avvio effettivo dello sviluppo e richiedere formazione aggiuntiva.

La struttura del sistema sarà divisa in servizi indipendenti per il Back End e il Front End.

Servizio Back End (Implementato in Python e Flask):



Il Back End gestirà la logica applicativa, la gestione delle richieste del Front End, l'elaborazione dei dati e la comunicazione con il database.

Servizio Front End (Sviluppato con il Framework Angular):

Il Front End si occuperà dell'interfaccia utente e dell'interazione diretta con gli utenti del sistema, utilizzando i servizi Back End per ottenere e inviare dati.

Servizio di Persistenza dei Dati (Sviluppato con MySQL):

Per quanto riguarda la persistenza dei dati, il sistema si baserà su MySQL come database relazionale, integrato nei servizi attraverso l'Object-Relational Mapping (ORM) SQLAlchemy. Questa scelta mira a garantire un'efficace gestione dei dati, consentendo un accesso sicuro e performante alle informazioni archiviate.

La comunicazione tra il frontend e il backend avviene mediante protocollo HTTP. Il frontend, sviluppato con Angular, utilizza il modulo HttpClient per inviare richieste HTTP al backend, implementato in Python con Flask.

Il backend espone diverse API (Application Programming Interface) attraverso endpoint HTTP, ciascuno associato a una funzionalità specifica.

Il frontend invia richieste HTTP a questi endpoint specifici per ottenere dati o eseguire azioni sul server. Il backend processa queste richieste, interagisce con i servizi interni, elabora i dati e restituisce le risposte al frontend. Questa comunicazione avviene attraverso uno scambio strutturato di dati, nel formato JSON.

L'utilizzo di richieste HTTP standard facilita l'interoperabilità tra il frontend e il backend, consentendo una comunicazione efficace e la trasmissione di informazioni essenziali per il corretto funzionamento dell'applicazione.

3.2. Decomposizione in sottosistemi

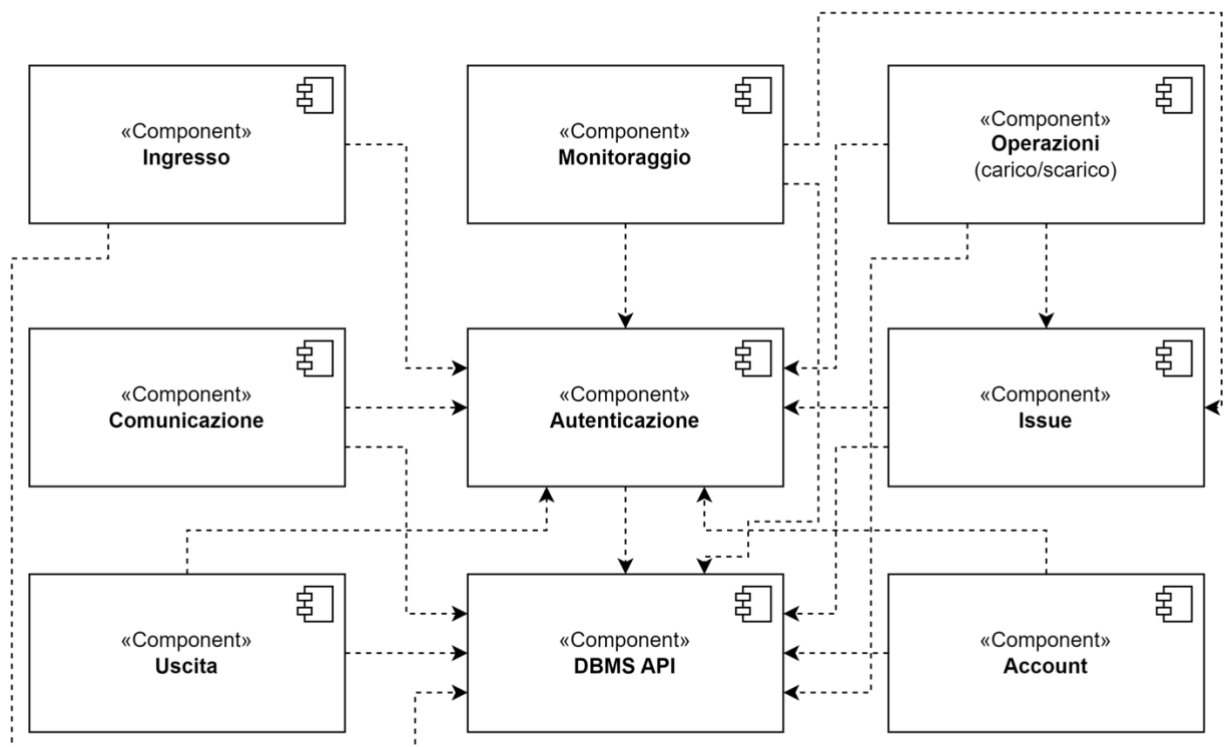
I sottosistemi individuati sono:

- **Ingresso:** si occupa di gestire le procedure di ingresso all'interno del porto.
- **Monitoraggio:** si occupa di tenere traccia dei movimenti dei camion all'interno del porto.
- **Comunicazione:** si occupa di gestire la comunicazione tra gli utenti abilitati.



- **Operazioni di Carico e Scarico:** si occupa di gestire lo svolgimento delle operazioni di carico e scarico all'interno del porto.
- **Issue:** si occupa di gestire le anomalie e la loro risoluzione.
- **Uscita:** gestisce il processo di uscita degli autotrasportatori dal porto.
- **Account:** si occupa della gestione degli account e dei relativi permessi per i diversi attori del sistema e della visualizzazione area utente.
- **Autenticazione:** si occupa delle operazioni di Login e Logout.
- **Persistenza:** gestisce la memorizzazione dei dati utilizzando un database.
- **DBMS API:** funge da intermediario tra i vari sottosistemi e il sottosistema di Persistenza.

Di seguito vengono mostrate le dipendenze tra i sottosistemi attraverso un component diagram UML.



Alcuni sottosistemi saranno gestiti utilizzando componenti COTS (Commercial off the Shelf). Ecco un elenco di tali componenti e una breve descrizione di ciascun sottosistema:

- Storage sarà gestito attraverso Flask, un framework leggero per applicazioni web in Python che fornisce funzionalità per la gestione del routing e l'interfacciamento con il database. Sarà utilizzato



Laurea Triennale in informatica - Università di Salerno
Corso di *Ingegneria del Software* -
Prof.ssa F. Ferrucci & Prof. F. Palomba

per gestire il livello di persistenza dei dati utilizzando SQLAlchemy, un ORM (Object-Relational Mapping) per Python che facilita l'interazione con il database.

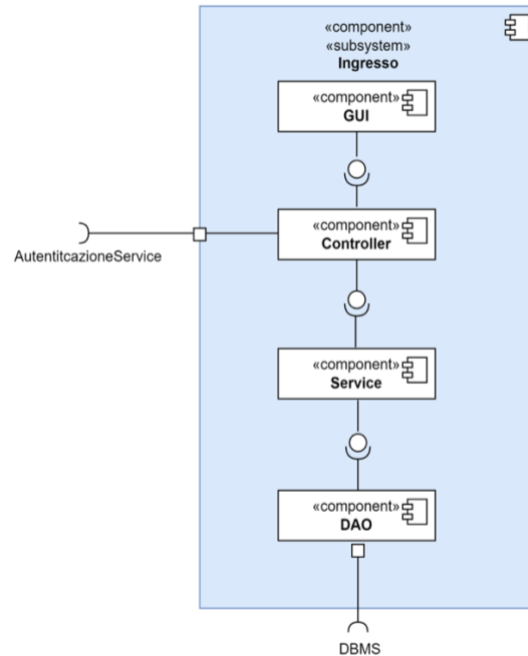
- Persistenza sarà gestita tramite un DBMS relazionale come MySQL, utilizzato per la memorizzazione persistente dei dati. SQLAlchemy fornisce un'interfaccia per comunicare con il DBMS relazionale, consentendo di creare, modificare e interrogare tabelle relazionali per l'archiviazione e il recupero dei dati.

Ogni sottosistema comprende le seguenti componenti principali:

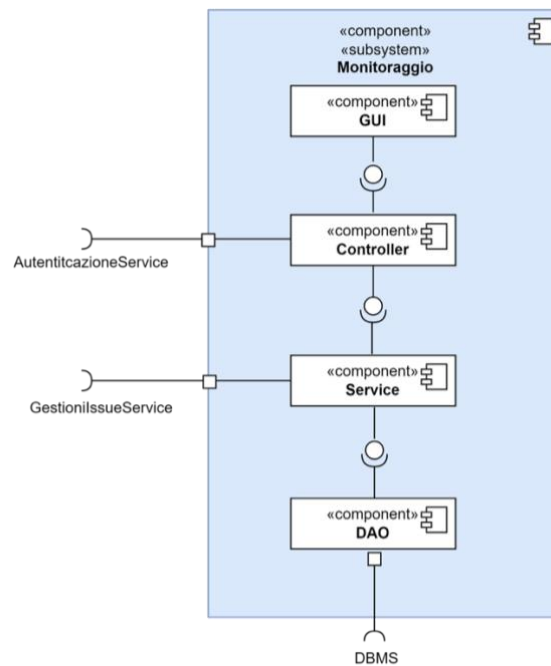
- **GUI** (Graphic User Interface): rappresenta le diverse visualizzazioni che verranno reindirizzate per creare le pagine web da mostrare all'utente. Queste view consentono agli utenti di interagire con il sistema in modo intuitivo.
- **Controller**: gestisce la logica per il controllo del sistema. Riceve le richieste degli utenti provenienti dalla GUI e le indirizza alle componenti appropriate per un'ulteriore elaborazione.
- **Service**: gestisce la logica di business del sistema. Qui vengono implementate le regole e le operazioni che definiscono il comportamento del sistema. Il Service si occupa di elaborare le richieste ricevute dal Controller e di eseguire le operazioni necessarie per soddisfare tali richieste.
- **DAO** (Data Access Object): si occupa di fornire un'interfaccia per l'accesso ai dati persistenti. Il DAO utilizza SQLAlchemy per interagire con il DBMS relazionale, definendo metodi per l'inserimento, l'aggiornamento, la cancellazione e la ricerca dei dati.



Sottosistema Ingresso

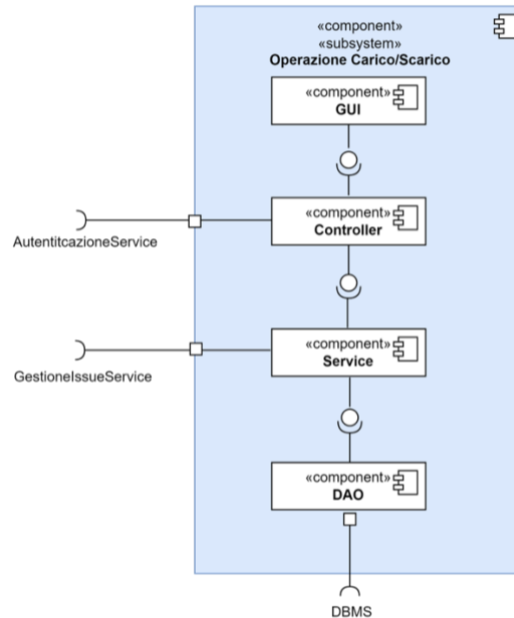


Sottosistema Monitoraggio

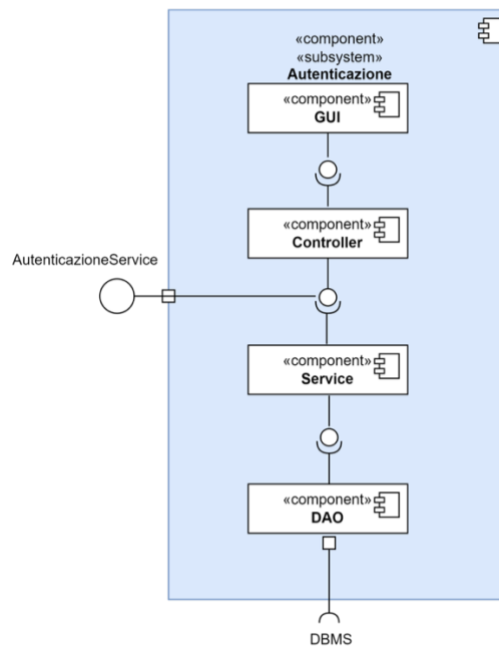




Sottosistema Operazione carico/scarico

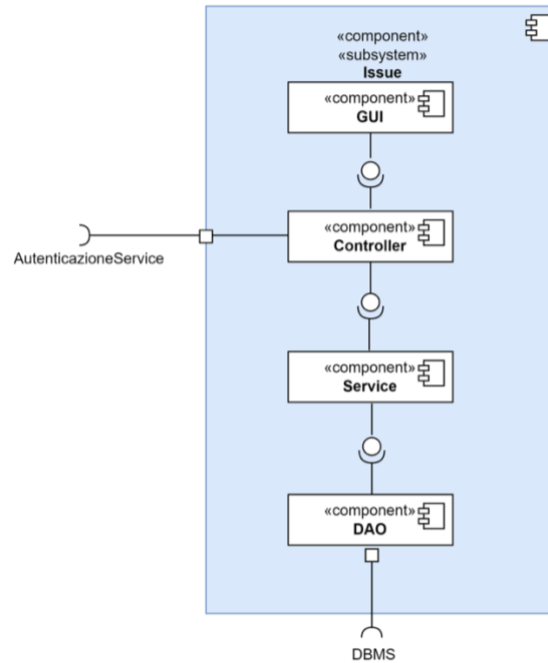


Sottosistema Autenticazione

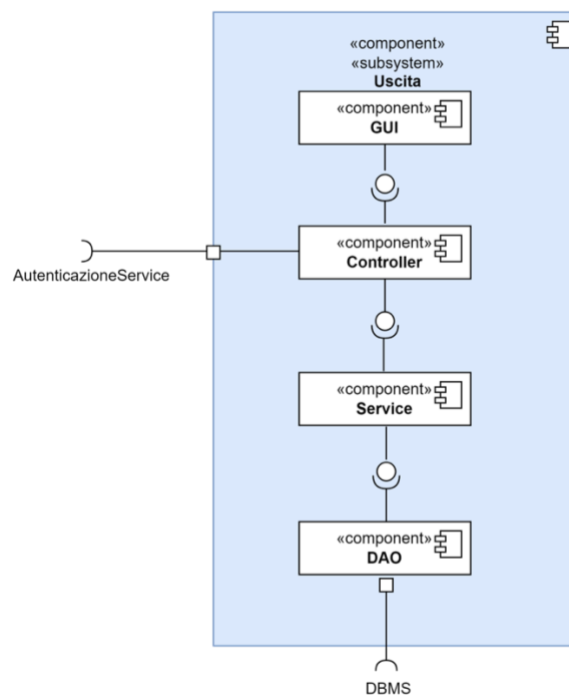




Sottosistema Issue

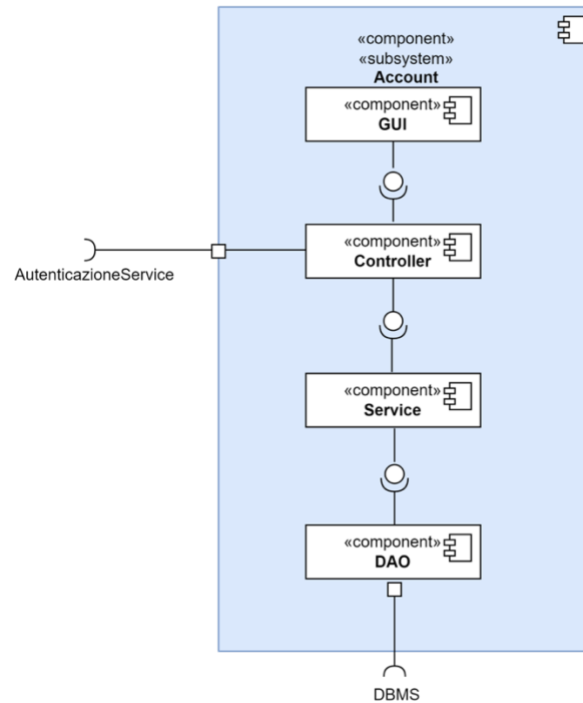


Sottosistema Uscita

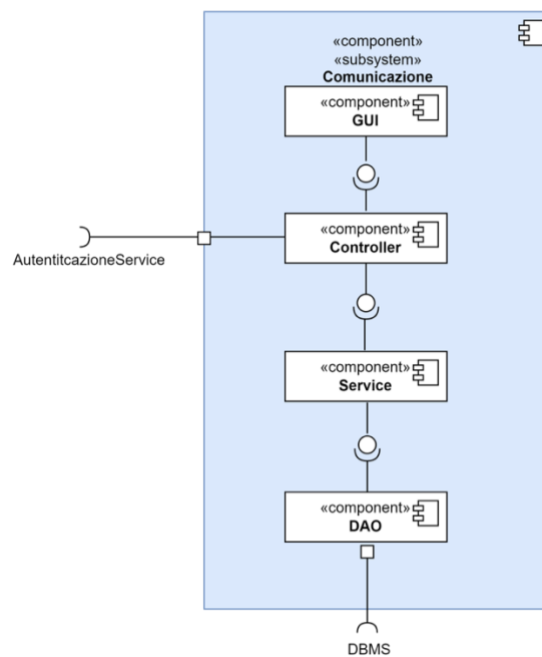




Sottosistema Account



Sottosistema comunicazione





L'applicazione web che stiamo sviluppando si appoggia su un server hardware dedicato, il quale gestisce le richieste provenienti dai diversi client degli utenti connessi.

Il diagramma mostra chiaramente come il software e l'hardware sono mappati, evidenziando la connessione tra il server e i client che interagiscono tramite il browser, semplificando la gestione e il controllo delle risorse, ma richiede una valutazione attenta delle capacità del server per evitare possibili limitazioni di scalabilità e rischi di single point of failure.





3.4. Gestione dei dati persistenti

Per garantire la gestione efficiente e coerente dei dati persistenti del sistema è stata adottata l'implementazione di un database relazionale con il supporto di un DBMS. Questa scelta è motivata dalla necessità di gestire con successo l'accesso concorrente ai dati, assicurando al contempo la coerenza e l'integrità dei dati stessi.

Per interagire con il DBMS relazionale, sarà utilizzato SQLAlchemy, un framework Python che fornisce un'interfaccia versatile per la comunicazione con il database.

Attraverso SQLAlchemy, sarà possibile gestire in modo efficiente la creazione, la modifica e l'interrogazione delle tabelle relazionali all'interno del database MySQL. Questa interfaccia semplifica notevolmente le operazioni di archiviazione e recupero dei dati, consentendo una gestione robusta e flessibile della persistenza nel contesto del sistema.

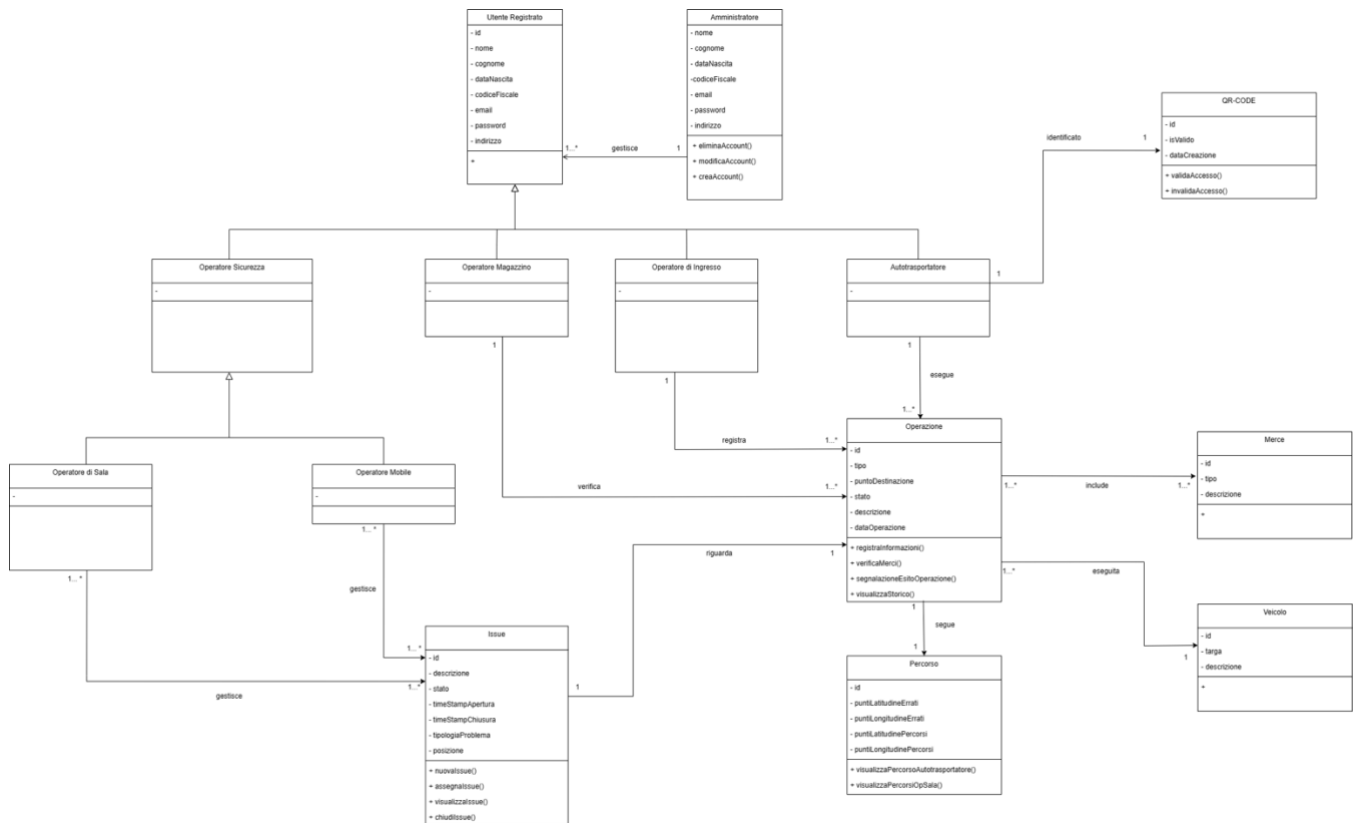
La combinazione SQLAlchemy e MySQL offre un ambiente coerente e potente per la gestione della persistenza, allineandosi agli standard di pratiche consigliate per lo sviluppo di sistemi complessi basati su database relazionali.

Sebbene i database relazionali siano ampiamente utilizzati e offrano coerenza e integrità dei dati, è importante considerare attentamente le sfide potenziali, specialmente in termini di:

- **Overhead di Query Complesse**, in quanto query complesse e join possono comportare un overhead in termini di prestazioni.
- **Rigidità dello Schema**, poiché gli schemi dei database relazionali sono strutturati e richiedono modifiche dello schema quando si aggiungono nuove entità o attributi.

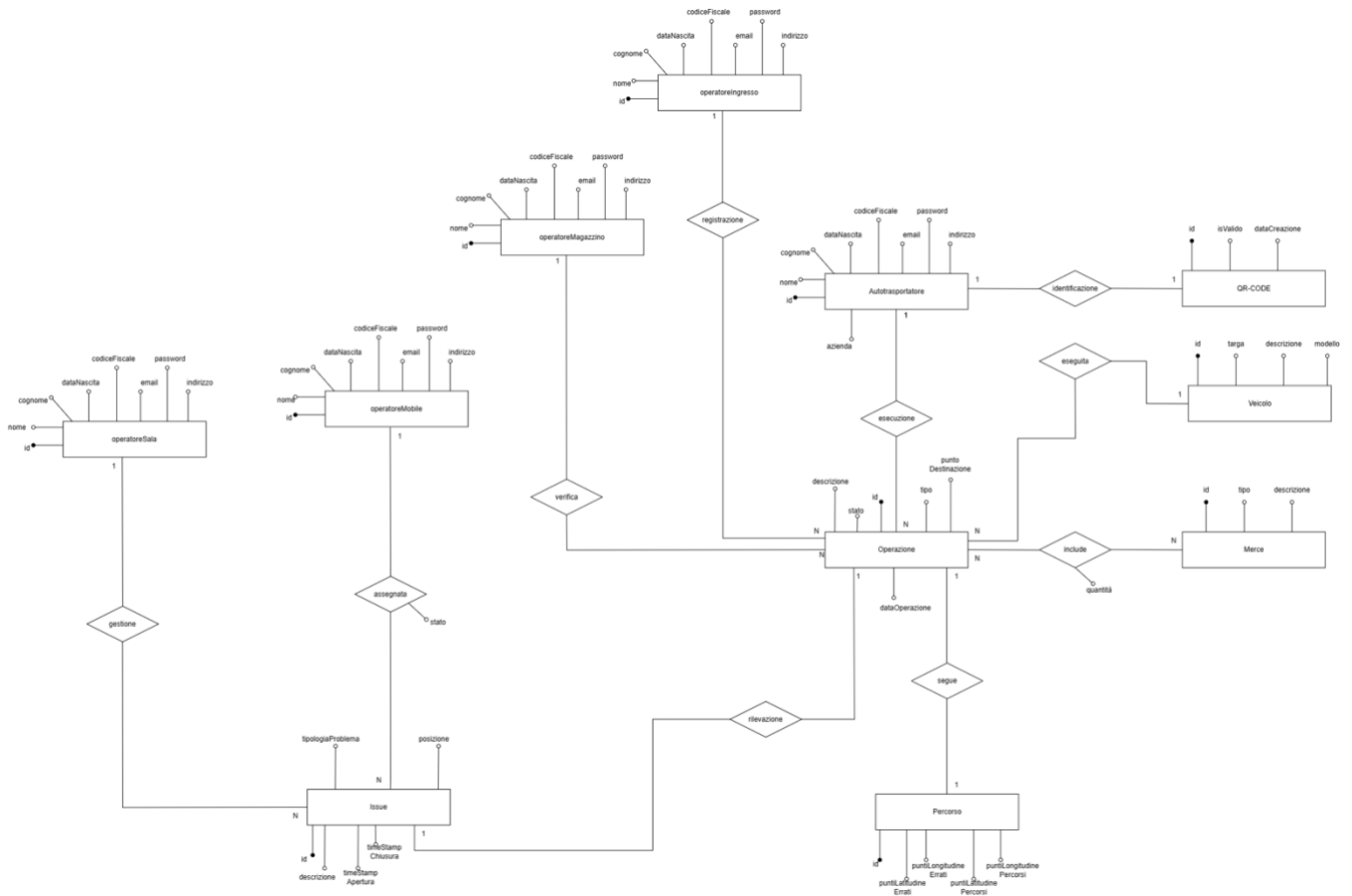


Entity Class Diagram – Autori: Mariapia Sorrentino, Paolo Murino, Francesco F. Ambrosio





Schema ER ristrutturato – Autori: Tutti i T.M.





Dizionario dei dati

Autotrasportatore

Nome Entità Autotrasportatore			
Descrizione	Contiene i Dati relativi all'autotrasportatore		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
nome	Varchar(30)		NOT NULL
cognome	Varchar(30)		NOT NULL
azienda	Varchar(255)		NOT NULL
dataNascita	Date		NOT NULL
codiceFiscale	Varchar(16)		NOT NULL
email	Varchar(255)		NOT NULL
password	Byte (32)		NOT NULL
indirizzo	Varchar(255)		NOT NULL
qrcode_id	Int (32)	FOREIGN KEY	NOT NULL

Veicolo

Nome Entità Veicolo			
Descrizione	Contiene i Dati relativi al veicolo		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
targa	Varchar(7)		NOT NULL
descrizione	Varchar(255)		
modello	Varchar(32)		NOT NULL



QR-CODE

Nome Entità	QR-CODE		
Descrizione	Contiene i Dati relativi al QR-CODE		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
isValido	Boolean		NOT NULL
dataCreazione	Date/Time		NOT NULL

Operatore Ingresso

Nome Entità	operatoreIngresso		
Descrizione	Contiene i Dati relativi all'operatore di ingresso		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
nome	Varchar(30)		NOT NULL
cognome	Varchar(30)		NOT NULL
dataNascita	Date		NOT NULL
codiceFiscale	Varchar(30)		NOT NULL
email	Varchar(255)		NOT NULL
password	Byte (32)		NOT NULL
indirizzo	Varchar(255)		NOT NULL



Operatore Magazzino

Nome Entità operatoreMagazzino			
Descrizione	Contiene i Dati relativi all'operatore di Magazzino		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
nome	Varchar(30)		NOT NULL
cognome	Varchar(30)		NOT NULL
dataNascita	Date		NOT NULL
codiceFiscale	Varchar(30)		NOT NULL
email	Varchar(255)		NOT NULL
password	Byte (32)		NOT NULL
indirizzo	Varchar(255)		NOT NULL

Operatore Mobile

Nome Entità operatoreMobile			
Descrizione	Contiene i Dati relativi all'operatore mobile		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
nome	Varchar(30)		NOT NULL
cognome	Varchar(30)		NOT NULL
dataNascita	Date		NOT NULL
codiceFiscale	Varchar(30)		NOT NULL
email	Varchar(255)		NOT NULL
password	Byte (32)		NOT NULL



indirizzo	Varchar(255)		NOT NULL
-----------	--------------	--	----------

Operatore Sala

Nome Entità operatoreSala			
Descrizione	Contiene i Dati relativi all'operatore mobile		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
nome	Varchar(30)		NOT NULL
cognome	Varchar(30)		NOT NULL
dataNascita	Date		NOT NULL
codiceFiscale	Varchar(30)		NOT NULL
email	Varchar(255)		NOT NULL
password	Byte (32)		NOT NULL
indirizzo	Varchar(255)		NOT NULL

Issue

Nome Entità Issue			
Descrizione	Contiene i Dati relativi all'issue		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
descrizione	Varchar(255)		NOT NULL
timestampApertura	DateTime		NOT NULL
timestampChiusura	DateTime		
stato	Varchar(255)		NOT NULL



tipologiaproblema	Varchar(255)		NOT NULL
posizione	Varchar(255)		NOT NULL
operatoresala_id	Int (32)	FOREIGN KEY	NOT NULL
operatoremobile_id	Int (32)	FOREIGN KEY	
operazione_id	Int (32)	FOREIGN KEY	NOT NULL

Operazione

Nome Entità	Operazione		
Descrizione	Contiene i Dati relativi all'operazione		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
dataOperazione	Date		NOT NULL
tipo	Varchar(32)		NOT NULL
descrizione	Varchar(255)		
puntoDestinazione	Varchar(32)		NOT NULL
stato	Varchar(32)		NOT NULL
autotrasportatore_id	Int (32)	FOREIGN KEY	NOT NULL
operatoreingresso_id	Int (32)	FOREIGN KEY	NOT NULL
operatoremagazzino_id	Int (32)	FOREIGN KEY	NOT NULL
percorso_id	Int (32)	FOREIGN KEY	NOT NULL
veicolo_id	Int (32)	FOREIGN KEY	NOT NULL



Include

Nome Entità	Include		
Descrizione	Contiene i Dati relativi alla corrispondenza tra operazione e merce		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
operazione_id	Int (32)	FOREIGN KEY	NOT NULL
merce_id	Int (32)	FOREIGN KEY	NOT NULL
quantità	double		NOT NULL

Merce

Nome Entità	Merce		
Descrizione	Contiene i Dati relativi alla corrispondenza tra operazione e merce		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
tipo	Varchar(32)		NOT NULL
descrizione	Varchar(255)		NOT NULL



Percorso

Nome Entità	Percorso		
Descrizione	Contiene i Dati relativi al percorso		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	Int (32)	PRIMARY KEY	NOT NULL AUTO_INCREMENT
puntiLatitudinePercorsi	Varchar(5000)		NOT NULL
puntiLongitudinePercorsi	Varchar(5000)		NOT NULL
puntiLatitudineErrati	Varchar(5000)		
puntiLongitudineErrati	Varchar(5000)		



3.5. Controllo degli accessi di sicurezza

Di seguito è presente una tabella che registra quali attori hanno autorizzazione ad accedere a specifici servizi forniti dal sistema.

Attori	Operatore di Ingresso	Operatore di Magazzino	Operatore Mobile	Operatore di Sala
Oggetti				
Autenticazione	Login, Logout	Login, Logout	Login, Logout	Login, Logout
Account	ModificaAccount, VisualizzaAreaUtente	ModificaAccount, VisualizzaAreaUtente	ModificaAccount, VisualizzaAreaUtente	ModificaAccount, VisualizzaAreaUtente
Ingresso	validaAccesso, InvalidaAccesso, registraInformazioni			
Monitoraggio				visualizzaPercorsi, visualizzaStorico
Operazioni di carico e scarico		VerificaMerci, SegnalazioneEsito		
Comunicazione			ApriComunicazione, ChiudiComunicazione	ApriComunicazione, ChiudiComunicazione
Issue			VisualizzaIssue, ChiudiIssue	NuovaIssue, AssegnaIssue, ChiudiIssue, ModificaIssue
Uscita				



Attori	Amministratore	Autotrasportatore
Oggetti		
Autenticazione	Login, Logout	Login, Logout,
Account	CreaAccount, ModificaAccount, EliminaAccount	CreaAccount, ModificaAccount, VisualizzaAreaUtente
Ingresso		
Monitoraggio		
Operazioni di carico e scarico		VisualizzaPercorso
Comunicazione		ApriComunicazione ChiudiComunicazione
Issue		
Uscita		

La definizione rigida delle autorizzazioni di accesso, sebbene offra un controllo iniziale per garantire la sicurezza, potrebbe presentare alcuni aspetti contrari:

- **Rigidità nell'Aggiornamento delle Autorizzazioni:** La necessità di aggiornare frequentemente le autorizzazioni può portare a complessità gestionali.
- **Rischio di Errori Umani nella Manutenzione:** L'utilizzo di una tabella rigida per gestire le autorizzazioni potrebbe aumentare il rischio di errori umani durante la manutenzione. Ad esempio, potrebbe verificarsi l'assegnazione involontaria di autorizzazioni errate a un attore o il mancato aggiornamento delle autorizzazioni di un utente, con possibili conseguenze indesiderate sulla sicurezza del sistema.



3.6. Controllo globale del software

Il sistema “Smart Cargo” è un sistema interattivo, col quale l’utente potrà interfacciarsi mediante l’applicazione web. La sua architettura si basa su un approccio orientato agli eventi, mirando a garantire una gestione dinamica e interattiva delle attività portuali. Quando un utente avvia l'applicazione web, può interagire con diverse funzionalità del sistema. Ad esempio, la pianificazione dei percorsi dei camion all'interno del porto è gestita attraverso un meccanismo in cui l'utente viene abilitato ad accedere al porto, innescando un evento che avvia il processo di definizione dei percorsi. Questo processo utilizza algoritmi avanzati per determinare i percorsi più efficienti, tenendo conto di vari fattori come il traffico portuale e la riduzione delle emissioni di CO₂. Durante il transito dei camion nel porto, il sistema offre una tracciatura in tempo reale, consentendo agli operatori di monitorare la posizione esatta dei veicoli. Questa funzione viene attivata attraverso un evento gestito da una specifica classe Handler, che si interfaccia con il sottosistema dedicato al tracciamento dei veicoli. Tale processo di tracciamento fornisce una panoramica immediata e aggiornata delle attività portuali.

In conclusione, l'architettura orientata agli eventi del sistema "Smart Cargo" offre un approccio flessibile e dinamico per potenziare le attività portuali. Attraverso la gestione degli eventi, il sistema si adatta in modo continuo alle esigenze degli operatori, contribuendo a migliorare l'efficienza, la sicurezza e la sostenibilità ambientale delle operazioni portuali nel Porto.



3.7. Condizioni limite

In questo paragrafo, esploreremo le condizioni di confine relative all'avvio, allo spegnimento e al fallimento del sistema, nonché agli errori di accesso ai dati persistenti.

Avvio del sistema

Identificativo UCBC_1	Avvio del sistema	Data	30/11/2023
		Vers.	1.0
		Autori	Paolo Murino Mariapia Sorrentino Andreea C. C. Oprisescu
Descrizione	Lo UC permette l'avvio del sistema		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'Amministratore accede al server		
Exit condition (On success)	Il sistema viene avviato correttamente		
Exit condition (On failure)	Il sistema non viene avviato		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore	Esegue sulla macchina il comando che avvia il sistema	
2	Sistema:	Controlla l'integrità dei dati persistenti e, se risultano validi, attiva i suoi servizi per renderli accessibili agli utenti, garantendo la disponibilità delle sue funzioni	
I Scenario/Flusso di eventi ALTERNATIVO: I Dati persistenti non son integri			
2.1	Sistema:	Segnala eventuali anomalie ai dati persistenti all'Amministratore senza procedere con l'avvio	
2.2	Amministratore:	Procede a correggere i dati che risultano danneggiati	
2.3	Amministratore:	Esegue il Passaggio 1	



Spegnimento

Identificativo UCBC_2	Spegnimento del sistema	Data	30/11/2023
		Vers.	1.0
		Autori	Paolo Murino Mariapia Sorrentino Andreea C. C. Oprisescu
Descrizione	Lo UC permette lo spegnimento del sistema		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'Amministratore accede al server AND Il sistema è attualmente in esecuzione		
Exit condition (On success)	Il sistema viene spento correttamente.		
Exit condition (On failure)	Il sistema non viene spento.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore	Esegue sulla macchina il comando che porta allo spegnimento del sistema.	
2	Sistema:	Controlla che non ci siano operazioni in corso e, se non ci sono, effettua lo spegnimento del sistema.	
I Scenario/Flusso di eventi ALTERNATIVO: Ci sono operazioni ancora in corso			
2.1	Sistema:	Segnala all'Amministratore che ci sono ancora operazioni in corso all'interno del corso.	
2.2	Sistema:	Attende una quantità di tempo per la terminazione delle operazioni in corso, non permettendo l'avvio di nuove operazioni.	
2.3	Sistema:	Controlla che non ci siano operazioni ancora in corso e, se non ci sono, effettua lo spegnimento del sistema.	
2.4	Sistema:	Notifica all'Amministratore l'avvenuto spegnimento del sistema.	



Fallimento del sistema

Identificativo UCBC_3	Fallimento del sistema	Data	30/11/2023
		Vers.	1.0
		Autori	Paolo Murino Mariapia Sorrentino Andreea C. C. Oprisescu
Descrizione	Lo UC descrive il comportamento del sistema nel caso esso fallisca.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema viene interrotto in modo imprevisto.		
Exit condition (On success)	Il sistema è riavviato con successo.		
Exit condition (On failure)	Il sistema non viene riavviato correttamente.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore	Include UCBC_1	



Errore di Accesso ai Dati Persistenti

Identificativo UCBC_4	Errore di Accesso ai Dati Persistenti	Data	30/11/2023
		Vers.	1.0
		Autori	Paolo Murino Mariapia Sorrentino Andreea C. C. Oprisescu
Descrizione	Lo UC descrive il comportamento del sistema nel caso in cui l'accesso ai dati persistenti risulti impossibile o se tali dati risultino corrotti.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema è impossibilitato ad accedere ai dati persistenti. OR I dati persistenti risultano essere corrotti		
Exit condition (On success)	Il sistema ritorna al normale funzionamento.		
Exit condition (On failure)	Il sistema non ritorna al normale funzionamento.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Il sistema notifica all'amministratore dell'incapacità di accedere ai dati persistenti.	
2	Sistema:	Il sistema interrompe il processo di avvio di nuove operazioni e restituisce un messaggio di errore alle operazioni precedentemente aperte.	
3	Amministratore:	Include UCBC_2	
4	Amministratore:	Ripristina l'accessibilità o l'integrità dei dati persistenti.	
5	Amministratore:	Include UCBC_1	



4. Servizi dei sottosistemi

Sottosistema autenticazione

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità permette di accedere al sistema tramite le proprie credenziali.	AutenticazioneController
Logout	Questa funzionalità permette di disconnettersi al sistema.	AutenticazioneController

Sottosistema ingresso

Servizio	Descrizione	Interfaccia
Validazione accesso	Questa funzionalità permette di validare il QRCODE consentendo l'accesso al porto.	IngressoController
Invalidazione accesso	Questa funzionalità permette di invalidare il QRCODE negando l'accesso al porto.	IngressoController
Registrazione informazioni	Questa funzionalità permette di registrare nel sistema le informazioni relative a un'operazione associata a un'autotrasportatore.	IngressoController

Sottosistema monitoraggio

Servizio	Descrizione	Interfaccia
Visualizza Percorsi	Questa funzionalità permette di monitorare i percorsi seguiti dagli autotrasportatori nel porto	MonitoraggioController
Visualizza Storico	Questa funzionalità permette l'accesso ad un registro storico delle operazioni dei camion, incluso il tracciamento delle rotte	MonitoraggioController



Sottosistema operazione carico/scarico

Servizio	Descrizione	Interfaccia
VerificaMerci	Questa funzionalità permette di verificare le merci.	OperazioniController
SegnalazioneEsito	Questa funzionalità permette di segnalare l'esito delle operazioni di carico/scarico.	OperazioniController

Sottosistema issue

Servizio	Descrizione	Interfaccia
NuovaIssue	Questa funzionalità permette di aprire un'issue relativa ad anomalie o problemi riguardanti il monitoraggio delle attività.	IssueController
VisualizzaIssue	Questa funzionalità permette di visualizzare un'issue relativa ad anomalie o problemi riguardanti il monitoraggio delle attività.	IssueController
ChiudiIssue	Questa funzionalità permette di chiudere un'issue relativa ad anomalie o problemi riguardanti il monitoraggio delle attività.	IssueController
AssegnaIssue	Questa funzionalità permette di assegnare un'issue ad un operatore Mobile.	IssueController
ModificaIssue	Questa funzionalità permette di modificare le informazioni relative ad un'issue.	IssueController



Sottosistema comunicazione

Servizio	Descrizione	Interfaccia
Apertura comunicazione	Questa funzionalità permette di aprire una comunicazione in caso di problemi.	ComunicazioneController
Chiusura comunicazione	Questa funzionalità permette di chiudere una comunicazione in caso di problemi.	ComunicazioneController

Sottosistema uscita

Servizio	Descrizione	Interfaccia
Uscita porto	Questa funzionalità permette la ad un autotrasportatore di completare l'uscita dal porto.	UscitaController

Sottosistema account

Servizio	Descrizione	Interfaccia
Creazione degli account	Questa funzionalità permette la creazione degli account per tutti gli attori del sistema	AccountController
Modifica degli account	Questa funzionalità permette la modifica dei dati dell'account per tutti gli attori del sistema	AccountController
Visualizzazione account	Questa funzionalità permette la visualizzazione del profilo personale	AccountController



5. Glossario

La seguente tabella fornisce una raccolta di definizioni chiare di termini chiave utilizzati nel documento.

Termine	Definizione
COTS	Commercial Off-The-Shelf", indica software, hardware o prodotti già pronti e disponibili sul mercato, pronti per l'acquisto e l'utilizzo immediato.
Flask	È un framework open source e gratuito, che consente agli sviluppatori di creare rapidamente e facilmente applicazioni web in Python.
Angular	Angular è un framework open-source sviluppato da Google, utilizzato per la creazione di applicazioni web dinamiche e single-page. Basato su TypeScript, Angular offre una struttura robusta e modulare per lo sviluppo front-end.
Classe Handler	Una classe handler è un componente software progettato per gestire eventi specifici all'interno di un sistema, attivando una funzione dedicata quando si verifica un evento correlato.
puntoDestinazione	Il punto di destinazione all'interno dell'area portuale può essere definito come una specifica localizzazione o posizione designata all'interno del porto, dove un'autorità o un operatore prevede che un veicolo o una merce debbano giungere o essere posizionati.
Stato	Si riferisce a una situazione o condizione particolare in cui si trova un'entità in un determinato momento.
Overhead	Indica la quantità di risorse aggiuntive (come tempo di elaborazione o memoria) richieste dal sistema per eseguire funzioni di gestione, controllo o sicurezza.