



Laurea Triennale in informatica - Università di Salerno
Corso di Ingegneria del software - Prof. Gravino Carmine



MAGAZON LAB

Anno accademico 2024/25

Repository GitHub

TEAM MEMBERS

Battaglia Daniel



Vaiano Francesco



Gigante Ruben



INDICE

- Breve introduzione
- Funzionalità del progetto
- Requisiti non funzionali
- Diagrammi UML
- Testing
- Demo live con spiegazione
- Conclusioni

STATEMENT OF WORK

Situazione attuale



MagazON Lab è una start-up italiana situata in Campania che ha l'incarico di gestire la logistica del magazzino per conto di Amazon.

Ha identificato il bisogno di un software gestionale per migliorare l'organizzazione del magazzino, ottimizzare i flussi interni e garantire una gestione delle operazioni logistiche più fluida, senza necessità di aumentare il personale.



STATEMENT OF WORK

Obiettivi di business



L'obiettivo è quello di ottimizzare e monitorare i flussi di prodotti all'interno del magazzino, garantendo un controllo efficace delle operazioni quotidiane e offrendo un supporto logistico per migliorare la gestione e l'organizzazione interna.

Non meno importante, è l'aumento di efficienza, rapidità e usabilità che questo porta all'attività.



FUNZIONALITÀ

Aree di gestione

Si è deciso di suddividere il progetto in 7 aree di gestione:



**Gestione
Prodotti**



**Gestione
Categorie**



**Gestione
Logistica**



**Gestione
Utenti**



**Gestione
Notifiche**



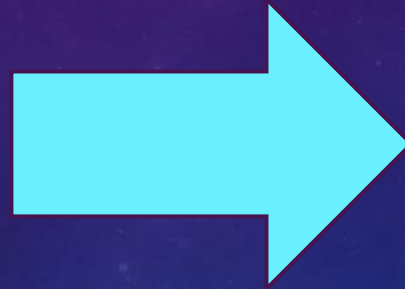
**Gestione
Liste**



**Gestione
Autenticazione**

FUNZIONALITÀ

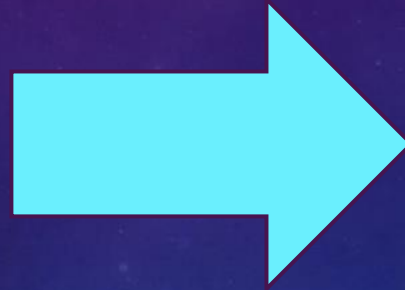
Gestione Prodotti



- Visualizzazione prodotti
- Inserimento prodotti
- Modifica prodotti
- Eliminazione prodotti

FUNZIONALITÀ

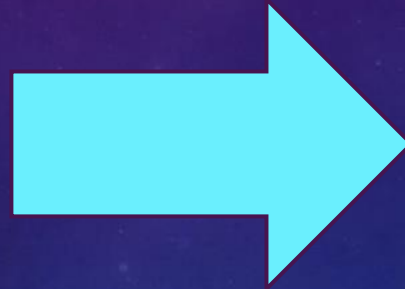
Gestione Categorie



- Visualizzazione categorie
- Inserimento categorie
- Modifica categorie
- Eliminazione categorie

FUNZIONALITÀ

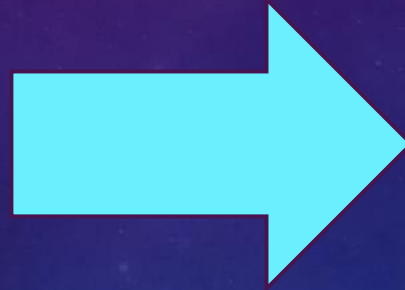
Gestione Logistica



- Visualizzazione arrivi e spedizioni
- Inserimento arrivi e spedizioni
- Modifica arrivi e spedizioni
- Eliminazione arrivi e spedizioni

FUNZIONALITÀ

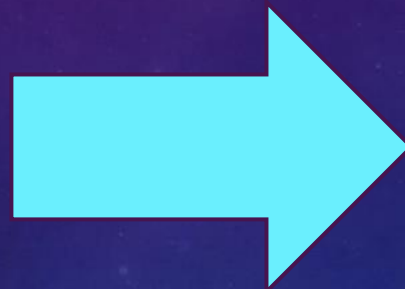
Gestione Utenti



- Visualizzazione account
- Inserimento account
- Modifica account
- Eliminazione account

FUNZIONALITÀ

Gestione Notifiche



- Invio notifica
- Visualizzazione notifiche

FUNZIONALITÀ

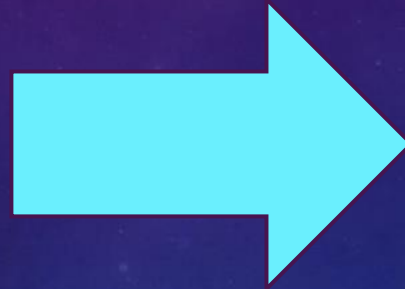
Gestione Liste



- Visualizzazione file lista
- Inserimento file lista
- Modifica file lista
- Eliminazione file lista

FUNZIONALITÀ

Gestione Autenticazione

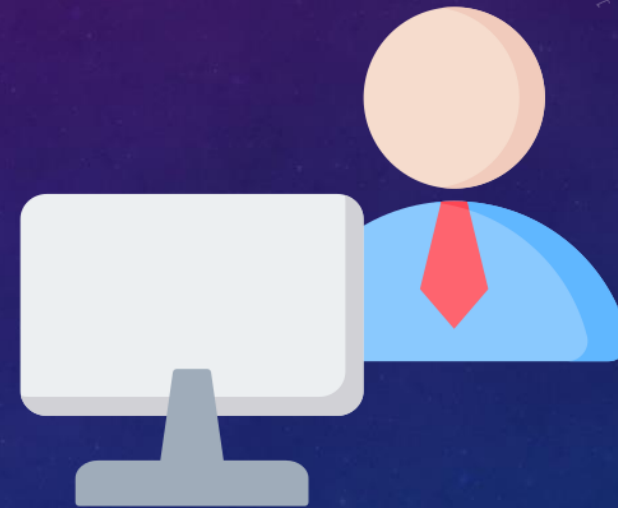


- Login
- Logout

DIFFERENZE TRA I RUOLI



Magazziniere



Amministratore

REQUISITI NON FUNZIONALI



PERFORMANCE



USABILITÀ



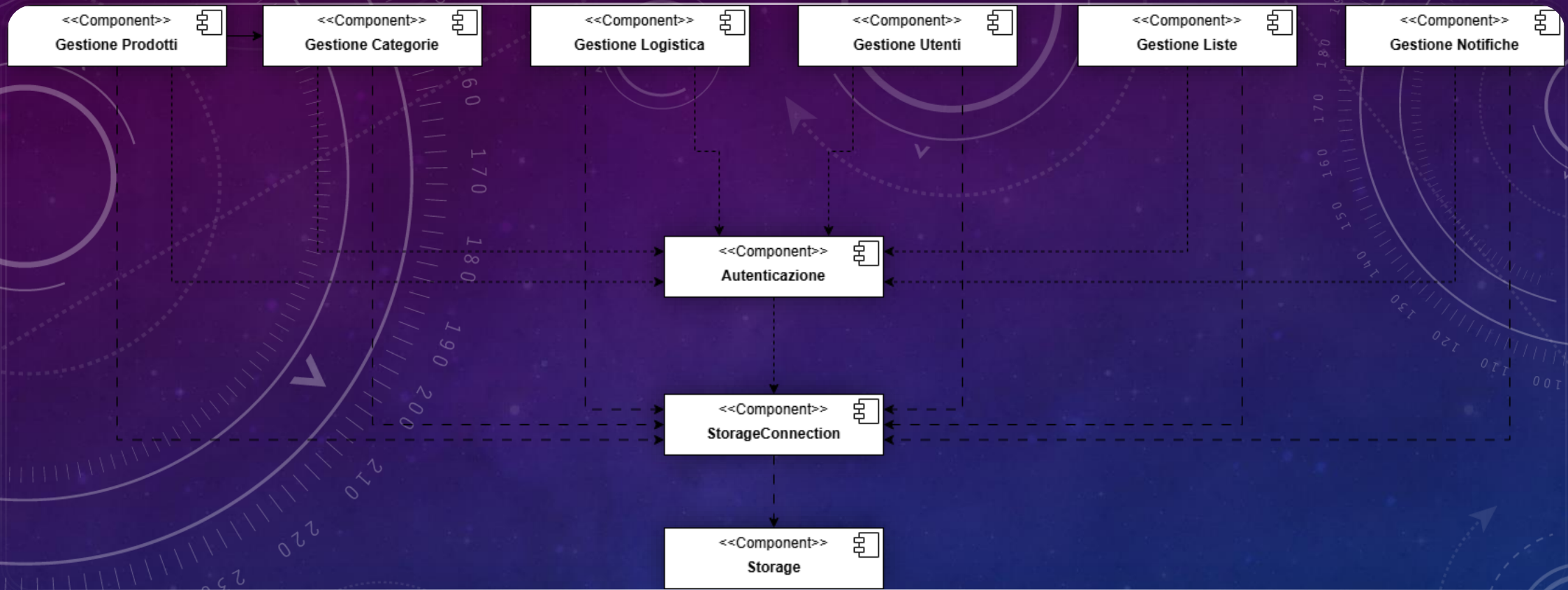
AFFIDABILITÀ



SUPPORTABILITÀ

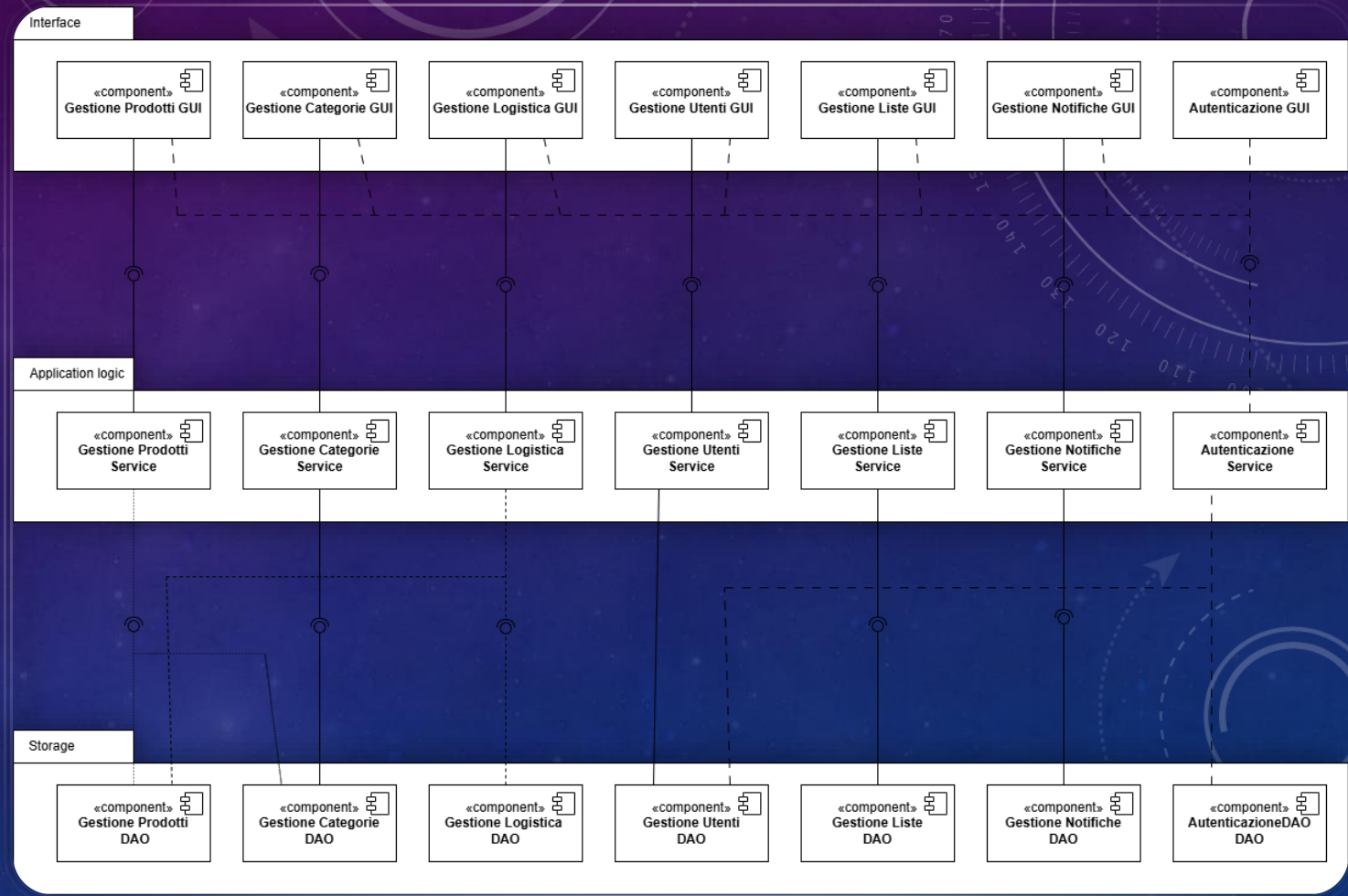
DESIGN GOALS E TRADE-OFF

Trade-off	Descrizione
Facilità d'uso vs. complessità	L'introduzione di funzionalità aggiuntive con una complessità ed un livello di dettaglio maggiori, accrescerebbe la completezza del sistema. Tuttavia, ciò renderebbe il sistema più difficile da utilizzare e da comprendere per gli utenti meno esperti.
Gestione dei fallimenti vs. tempi di risposta	Avere procedure complesse di gestione dei fallimenti potrebbe rallentare la risoluzione di problemi e influire sui tempi di risposta.
Riservatezza vs. condivisione dei dati	Effettuare una netta distinzione tra le operazioni permesse a ciascun utente, può rendere più difficile la comunicazione e la condivisione dei dati tra gli utenti del sistema.



DECOMPOSIZIONE IN SOTTOINSIEMI
COMPONENT DIAGRAM

ARCHITETTURA THREE TIER



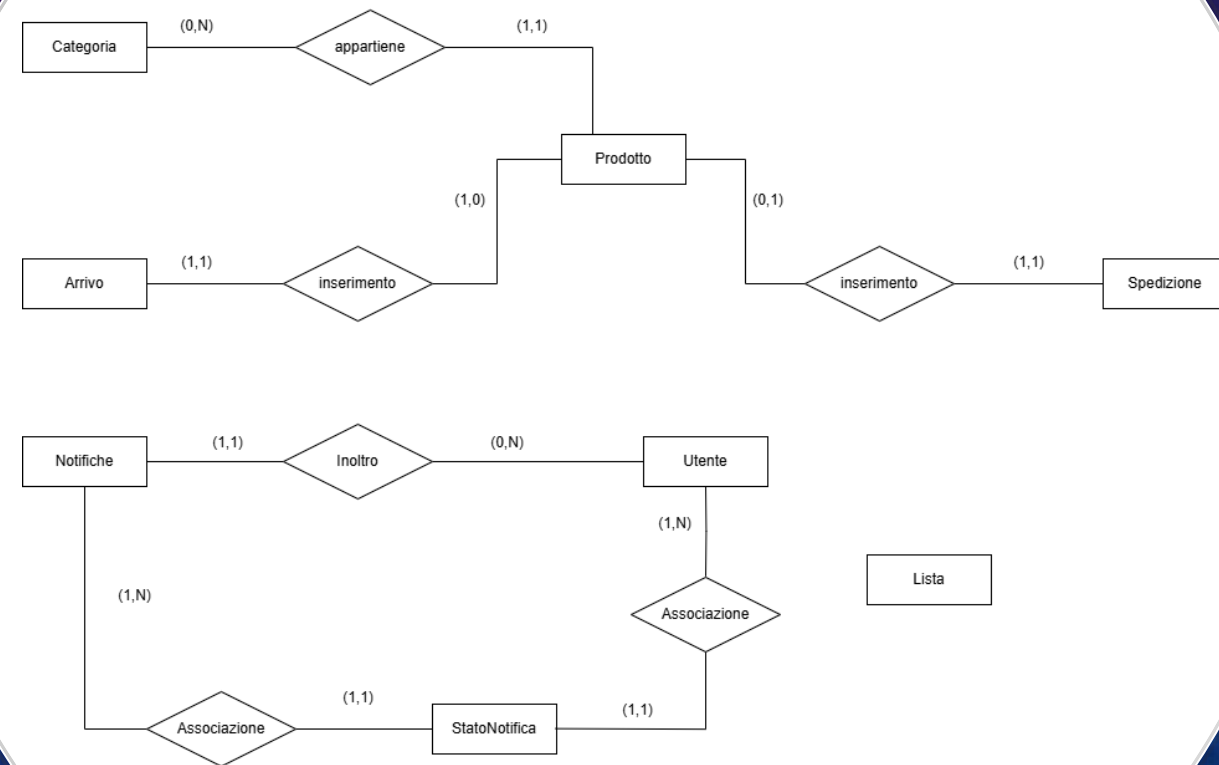
MAPPING HARDWARE-SOFTWARE DEPLOYMENT DIAGRAM

L'applicazione web che si intende realizzare sarà costituita da un server, risiedente su un solo nodo, che risponderà alle richieste di un client. Quest'ultimo, mediante l'interfaccia fornita dal browser, potrà effettuare operazioni col sistema mediante una richiesta HTTPS, alla quale il server procederà a rispondere con la funzionalità richiesta.

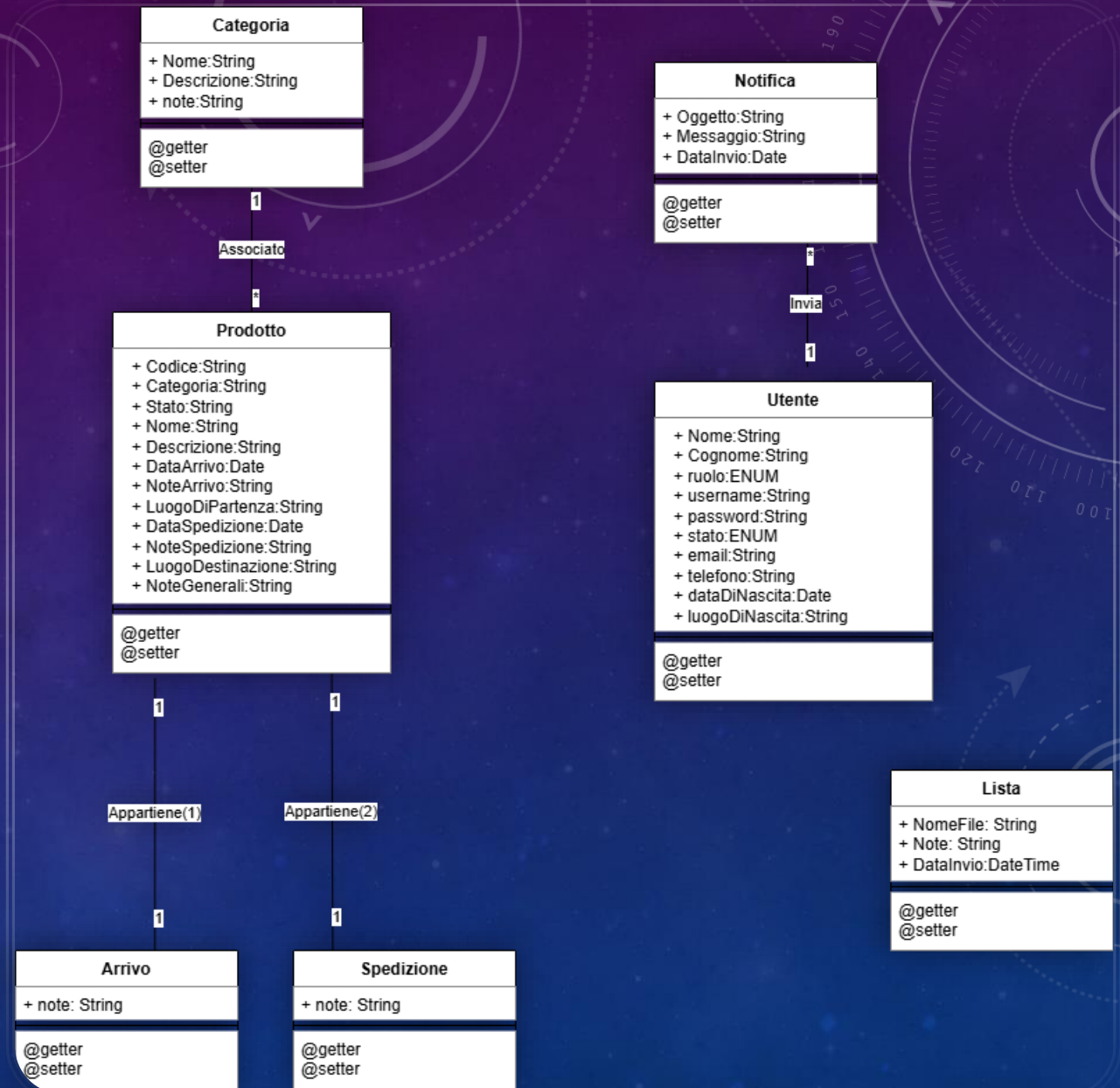
Il seguente **Deployment Diagram** descrive il mapping tra hardware e software.



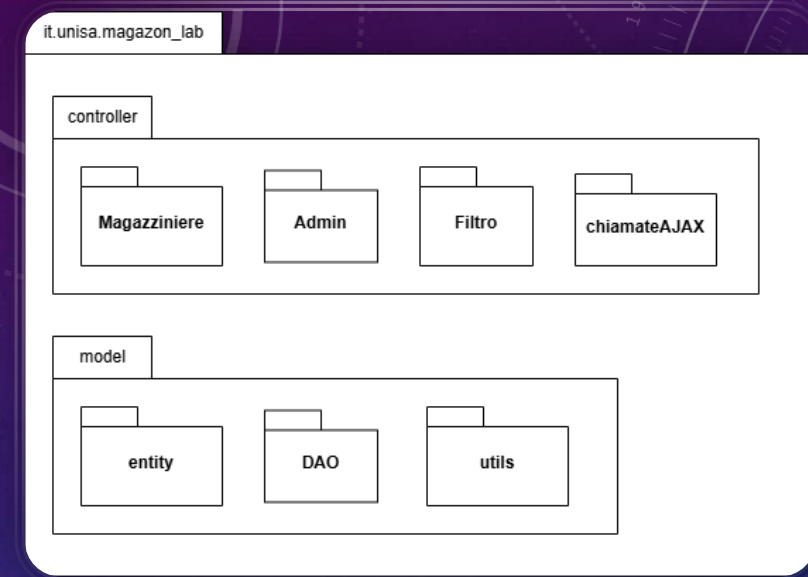
GESTIONE DEI DATI PERSISTENTI MODELLO ER



CLASS DIAGRAM RISTRUTTURATO



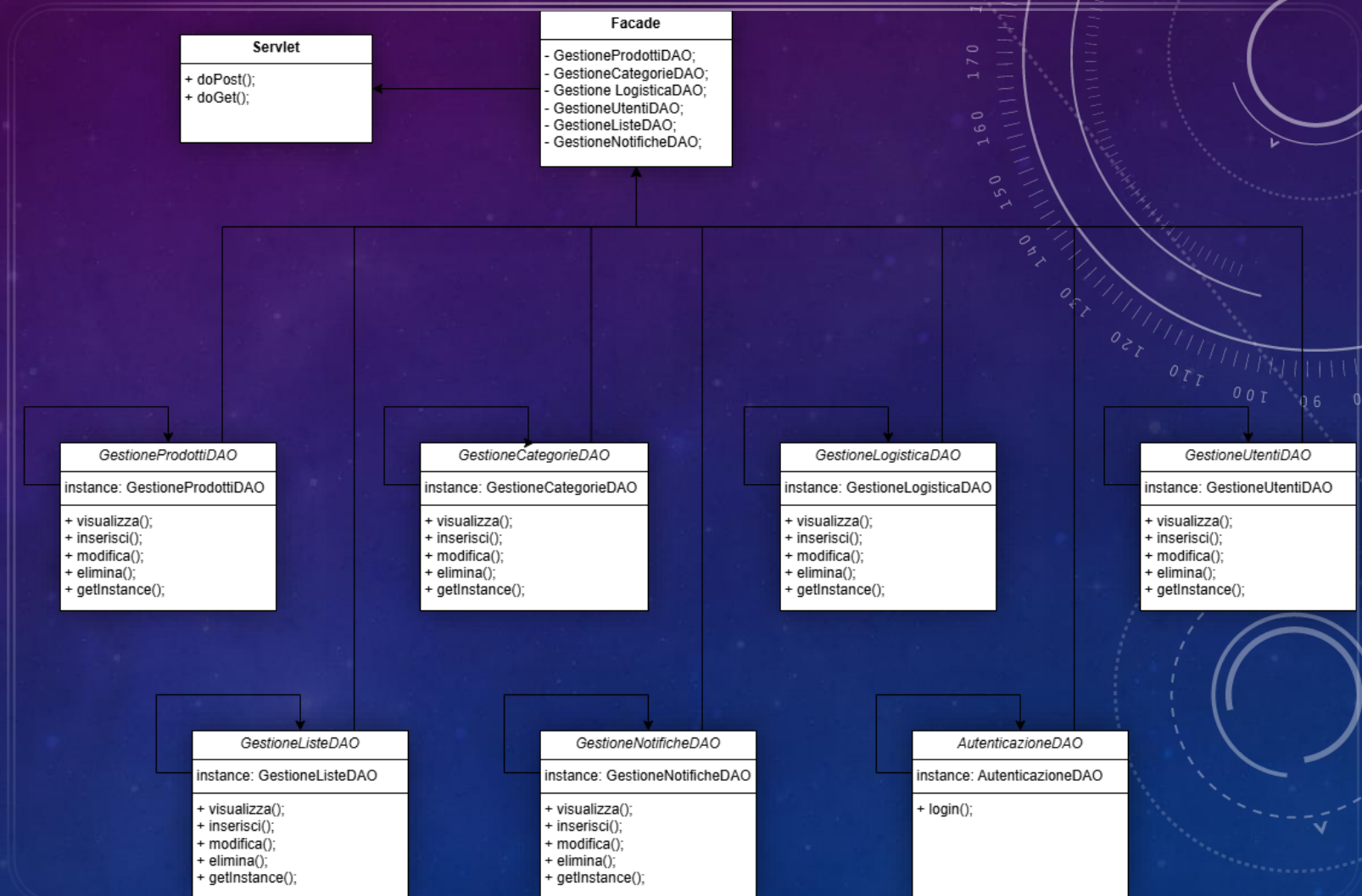
PACKAGE UML PACKAGE DIAGRAM



DESIGN PATTERN

FACADE

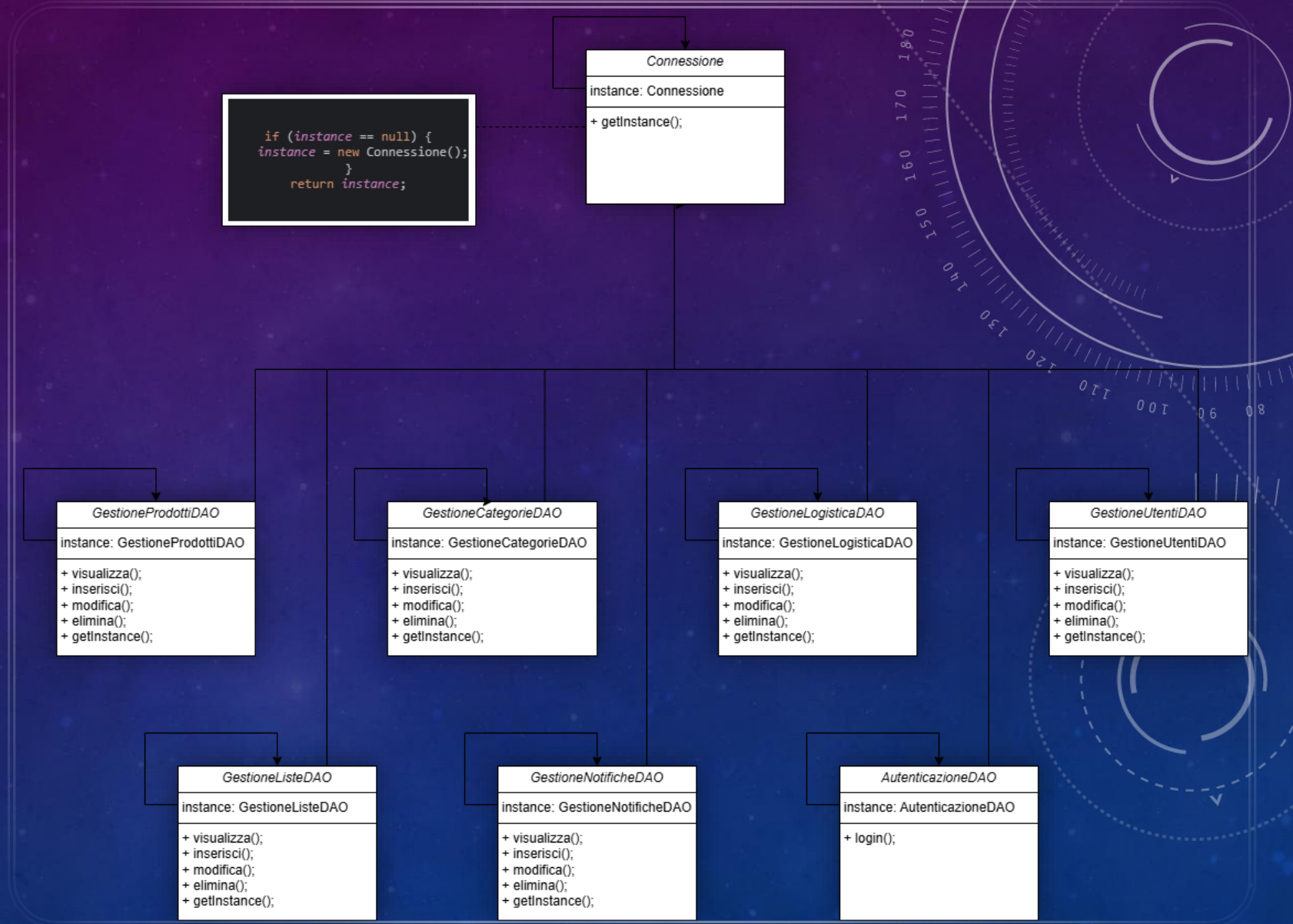
Utilizziamo una sola classe facade che fa da tramite con tutte le classi DAO, utili per le interrogazioni al database, e le servlet, essenziali per la gestione dei dati ricevuti in input e per l'invio di dati in output alla view. Quindi mettiamo a disposizione delle servlet la logica dei nostri sottosistemi, evitando che esse vadano ad interfacciarsi in modo diretto con le classi DAO.



DESIGN PATTERN

SINGLETON

Utilizziamo questo design pattern per gestire la connessione al database e le classi DAO, così da garantire che ogni classe che interagisce col database abbia una sola istanza.



TESTING

Panoramica

Testing di unità: utilizzo di JUnit per testare singoli metodi all'interno delle classi sviluppate.



Testing di integrazione: utilizzo di Mockito per testare l'integrazione tra componenti simulando il comportamento di quelle mancanti.

Testing di sistema: utilizzo di Selenium per simulare le interazioni dell'utente finale con il sistema.



Per i Test Cases abbiamo utilizzato il metodo **Category Partition**.

TESTING

Risultati TIR

Test Case ID	Test Incident ID	Tester	Data	Esito
TC_5.1.1	TC_5.1.1	RG	30/01/2025	Passed
TC_5.1.2	TC_5.1.2	RG	30/01/2025	Passed
TC_6.1.1	TC_6.1.1	DB	30/01/2025	Passed
TC_6.1.2	TC_6.1.2	DB	30/01/2025	Passed
TC_6.1.3	TC_6.1.3	DB	30/01/2025	Passed
TC_6.1.4	TC_6.1.4	DB	30/01/2025	Passed
TC_7.1.1	TC_7.1.1	FV	30/01/2025	Passed
TC_7.1.2	TC_7.1.2	FV	30/01/2025	Passed
TC_7.1.3	TC_7.1.3	FV	30/01/2025	Passed
TC_7.1.4	TC_7.1.4	RG	30/01/2025	Passed
TC_7.1.5	TC_7.1.5	RG	30/01/2025	Passed
TC_7.1.6	TC_7.1.6	RG	30/01/2025	Passed

Risultati

PASSED	FAILED	NON EFFETTUATI
45	0	0

Legenda

Cognome	Nome	Sigla
Battaglia	Daniel	DB
Vaiano	Francesco	FV
Gigante	Ruben	RG

TESTING

Integration testing

Per il **Testing di integrità** abbiamo scelto di verificare il funzionamento delle servlet per 3 metodi, uno per ogni membro del team.

Nome	Tester	Esito
InvioNotifica	DB	Passed
AggiungiUtente	RG	Passed
ModificaCategoria	FV	Passed

TESTING

Coverage

Testing Coverage: per analizzare la copertura del codice sorgente testato, abbiamo utilizzato JaCoCo.

JACOCO
Java Code Coverage

Questi sono i risultati principali:

Class Coverage	Method Coverage
39%	32%

ADESSO DIAMO UN'OCCHIATA AL SISTEMA

Ogni membro del team presenterà una funzionalità del sistema seguendo quest'ordine:

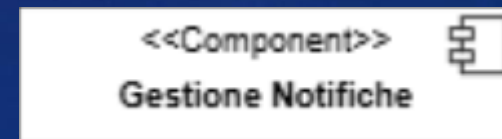
- Requisito Funzionale;
- Use Case;
- Eventuale Diagramma;
- Sottosistema Di Appartenenza;
- Demo Live

INVIO NOTIFICA

Requisito funzionale

ID	Nome	Descrizione	Attori	Priorità
RF_GN_1	Invio notifica	Il sistema deve permettere agli amministratori di poter inviare notifiche a tutti gli altri utenti.	Amministratore	ALTA

Invio notifica fa parte del component **Gestione Notifiche**



INVIO NOTIFICA

Interfaccia



Torna indietro

Invia notifica

*Oggetto:

*Messaggio:

Invia notifica



INVIO NOTIFICA

Use Case pt.1

Identificativo UC_GN1	Invio notifica	Data	23/12/2024
		Vers.	2.0
		Autore	Battaglia Daniel
Descrizione	Il seguente UC descrive l'invio di una notifica.		
Attore principale	Amministratore Interessato nell'invio di una notifica agli utenti.		
Attori secondari	NA		
Entry Condition	L'amministratore deve aver effettuato l'accesso al sistema dedicato agli amministratori.		
Exit condition On success	Viene mostrata correttamente la schermata per scrivere e inviare la notifica. La notifica viene inviata con successo.		

INVIO NOTIFICA

Use Case pt.2

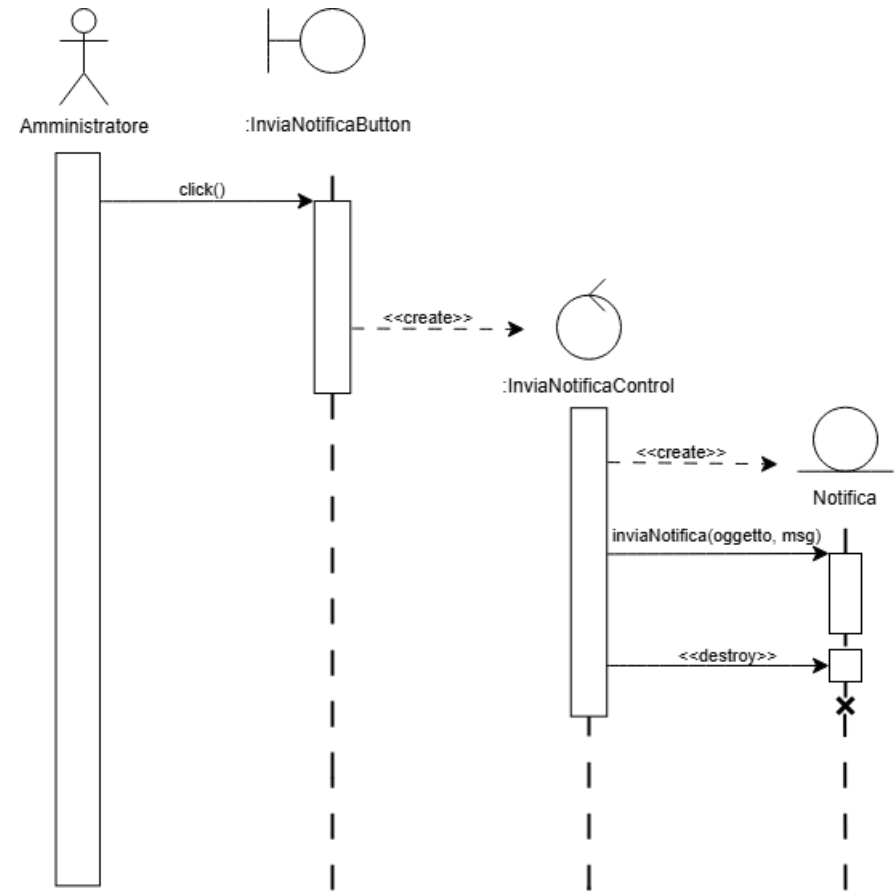
Exit condition On failure	Visualizzazione non eseguita correttamente o notifica non inviata.
Rilevanza/User Priority	Alta
Frequenza stimata	Da 1 a 15 notifiche al giorno
Extension point	NA
Generalization of	NA

INVIO NOTIFICA

Use Case pt.3

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Amministratore:	Dopo aver acceduto, l'amministratore accede all'area amministrativa per scrivere e inviare la notifica.
2	Sistema:	Il sistema mostra la schermata per la scrittura e l'invio della notifica.
I Scenario/Flusso di eventi di ERRORE : Il sistema non riesce ad inviare la notifica		
2.1	Sistema:	Il sistema mostra un messaggio di errore invio notifica
Note		
		NA
Special Requirements		
		NA

INVIO NOTIFICA SEQUENCE DIAGRAM





INVIIO NOTIFICA DEMO LIVE

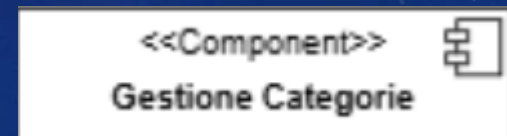
VEDIAMO COME UN AMMINISTRATORE INVIA UNA NOTIFICA!

VISUALIZZAZIONE CATEGORIE

Requisito funzionale

ID	Nome	Descrizione	Attori	Priorità
RF_GC_2	Visualizzazione e categorie	Il sistema dovrà permettere agli utenti di visualizzare tutte	Magazziniere/ Amministratore	ALTA

Visualizzazione categorie fa parte del component **Gestione Categorie**



VISUALIZZAZIONE CATEGORIE

Interfaccia

Aggiungi categoria

Torna indietro

LISTA CATEGORIE

ID	NOME	DESCRIZIONE	NOTE	AZIONE	
1	Cellulari345	Telefoni cellulari e non di casa		Modifica	Elimina
2	Arredamento	Articoli per la casa e arredamento	Include mobili e complementi d'arredo.	Modifica	Elimina
3	Abbigliamento	Vestiti e accessori per uomo, donna e bambino	Include calzature e accessori moda.	Modifica	Elimina
4	Cellulari	Telefono cellulari e non di casa		Modifica	Elimina

VISUALIZZAZIONE CATEGORIE

Use Case pt.1

Identificativo UC_GC1	Visualizzazione categorie	Data	23/12/2024
		Vers.	2.0
		Autore	Vaiano Francesco
Descrizione	Il seguente UC descrive la visualizzazione delle categorie.		
Attore principale	Magazziniere interessato a visualizzare le categorie.		
Attori secondari	NA		
Entry Condition	Il magazziniere deve aver effettuato l'accesso al sistema.		

VISUALIZZAZIONE CATEGORIE

Use Case pt.2

Exit condition On success	Viene mostrata correttamente la lista completa delle categorie inserite nel gestionale.
Exit condition On failure	Visualizzazione non eseguita correttamente.
Rilevanza/User Priority	Alta
Frequenza stimata	Dalle 0 alle 15 al giorno
Extension point	NA
Generalization of	NA

VISUALIZZAZIONE CATEGORIE

Use Case pt.3

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Magazziniere:	Dopo aver acceduto, il magazziniere accede all'area dedicata alle categorie.
2	Sistema:	Il sistema mostra l'elenco di tutte le categorie, dove ogni riga contiene le informazioni di una singola categoria.
I Scenario/Flusso di eventi di ERRORE : Il sistema non riesce a visualizzare le categorie		
2.1	Sistema:	Il sistema mostra l'errore e chiede all'utente di ricaricare la pagina.
Note		
		NA
Special <u>Requirements</u>		
		NA



VISUALIZZAZIONE CATEGORIE DEMO LIVE

VEDIAMO COME VISUALIZZARE LE CATEGORIE!

INSERIMENTO ACCOUNT


Requisito funzionale

ID	Nome	Descrizione	Attori	Priorità
RF_GU_1	Inserimento account	Il sistema dovrà permettere agli amministratori di inserire nuovi account magazzinieri, ma non amministratori.	Amministratore	ALTA

Inserimento account fa parte del component **Gestione Utenti**

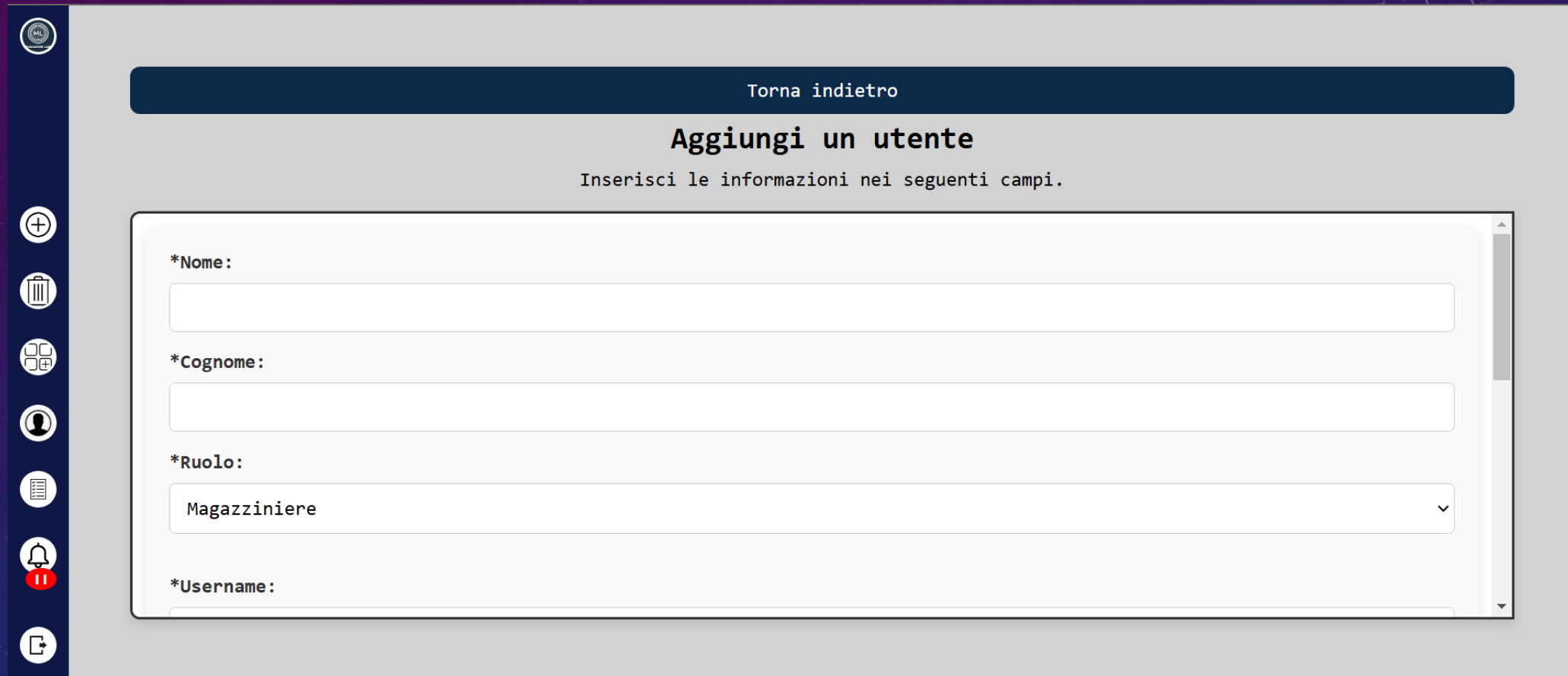
<<Component>>

Gestione Utenti



INSERIMENTO ACCOUNT

Interfaccia



The screenshot shows a web application interface for adding a new user. On the left is a dark blue sidebar with a vertical stack of icons: a circular logo at the top, followed by a plus sign, a trash can, a grid of four squares, a person silhouette, a document with lines, a bell with a red notification dot, and a document with a plus sign. The main content area has a light gray background. At the top of this area is a dark blue button labeled "Torna indietro". Below the button is the heading "Aggiungi un utente" in a bold, monospace font, followed by the instruction "Inserisci le informazioni nei seguenti campi." in a smaller monospace font. The form itself is a white box with a thin gray border and a vertical scrollbar on the right. It contains four labeled input fields: "*Nome:" with a text input, "*Cognome:" with a text input, "*Ruolo:" with a dropdown menu showing "Magazziniere" and a downward arrow, and "*Username:" with a text input.

Torna indietro

Aggiungi un utente

Inserisci le informazioni nei seguenti campi.

*Nome:

*Cognome:

*Ruolo:

*Username:

INSERIMENTO ACCOUNT

Use Case pt. 1

Identificativo UC_GU1	Registrazione account	Data	23/12/2024
		Vers.	2.0
		Autore	Gigante Ruben
Descrizione	Il seguente UC descrive la registrazione di un account.		
Attore principale	Amministratore Interessato a registrare nuovi dipendenti al sistema.		
Attori secondari	NA		
Entry Condition	L'amministratore deve aver effettuato l'accesso al sistema dedicato agli amministratori.		
Exit condition On success	Viene mostrata correttamente la schermata di registrazione e l'account viene registrato correttamente.		
Exit condition On failure	Visualizzazione del messaggio di registrazione fallita.		
Rilevanza/User Priority	Alta		
Frequenza stimata	Dai 10 ai 20 all'anno		
Extension point	NA		

INSERIMENTO ACCOUNT

Use Case pt. 2

<u>Generalization of</u>		NA
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Amministratore:	Dopo aver acceduto, l'amministratore accede all'area amministrativa per la registrazione di nuovi utenti.
2	Sistema:	Il sistema mostra il <u>form</u> di registrazione da compilare rispettando i requisiti.
I Scenario/Flusso di eventi di ERRORE : Il sistema non riesce a visualizzare i prodotti		
2.1	Sistema:	Il sistema mostra un messaggio di registrazione fallita e invita l'utente a riprovare.
Note		
		NA
<u>Special Requirements</u>		
		NA



INSERIMENTO ACCOUNT DEMO LIVE

VEDIAMO COME INSERIRE UN NUOVO MAGAZZINIERE

CONCLUSIONI

Cosa abbiamo imparato da questa esperienza?



Lavoro di squadra

Approccio allo sviluppo software



Ma soprattutto comunicazione con il team

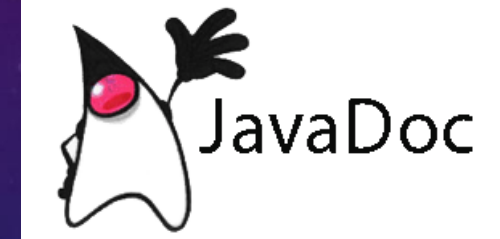
CONCLUSIONI

Strumenti per la comunicazione



CONCLUSIONI

Strumenti per lo sviluppo



Oltre quelli già utilizzati in altri corsi come: IntelliJ, Tomcat e MySQL



GRAZIE PER L'ATTENZIONE!