

Università di Bologna



Corso: Informatica per il Management

Ingegneria del Software

Anno Accademico: 2023/2024

Alex Ranaulo 1029929

Gioele Blanc 1019635

Alessandro Desolda 1020790

data consegna: **24/07/2024**

indice

1 Modello di dominio	2
2 UC Diagram.....	3
3. UC Testuali.....	3
4 Burn Down Chart.....	6
5 Manuale dell'utente.....	7
6 Manuale dello sviluppatore.....	12
7 Diario di Bordo.....	15

Progettazione Logica

Modelli di Dominio

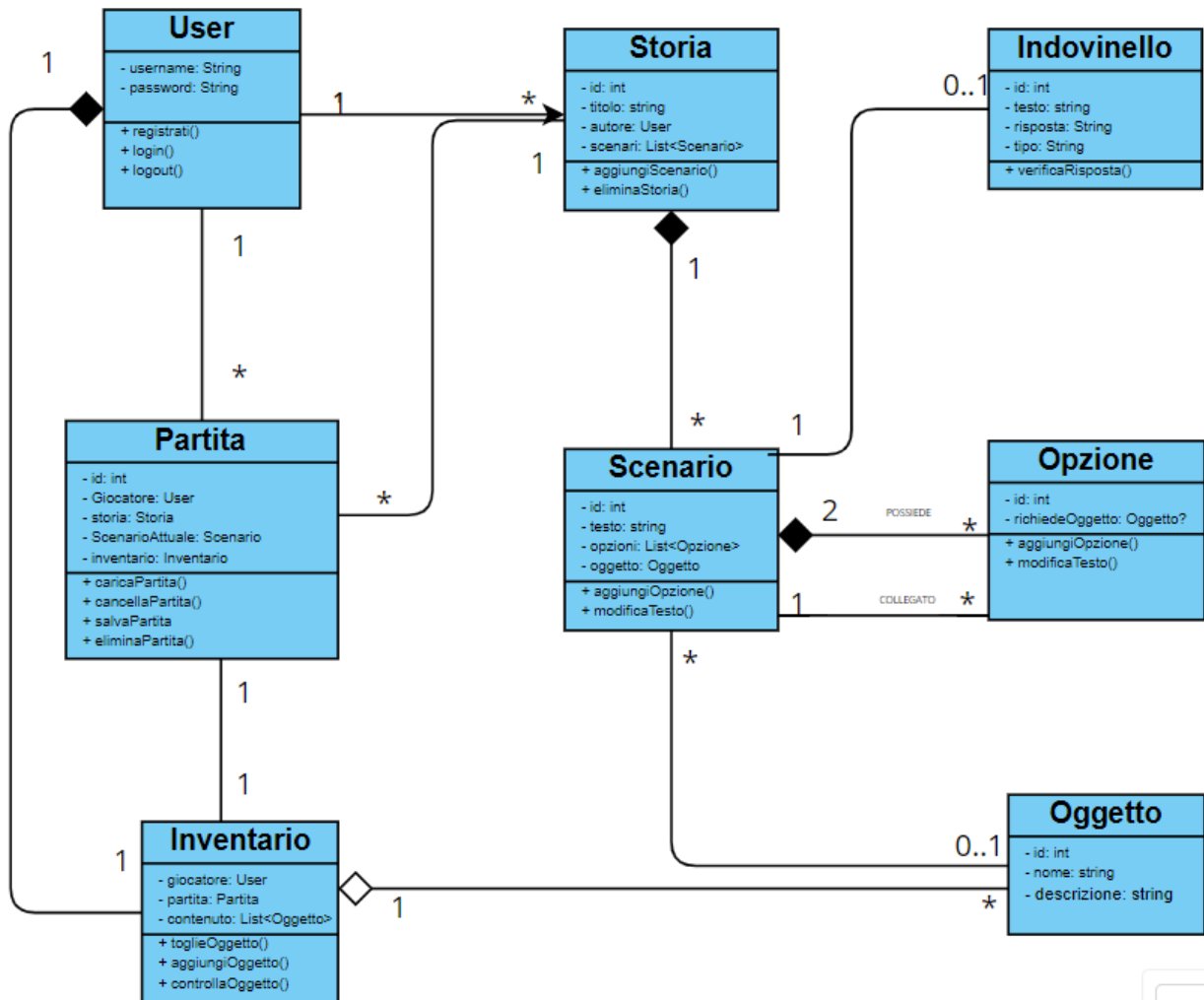
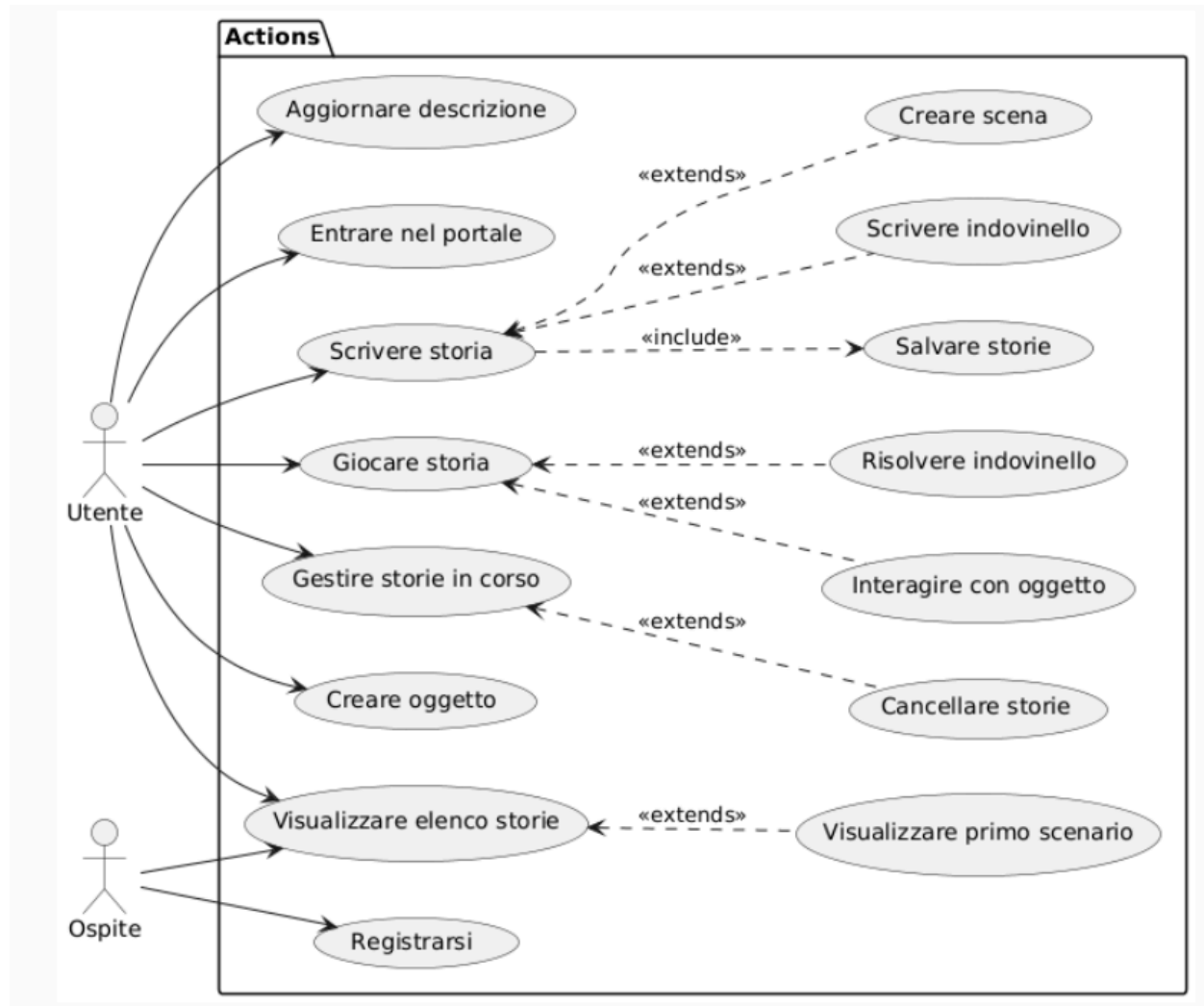


Diagramma Dei Casi d'Uso



Use Case testuali

Titolo	1.Entrare nel portale
Attori	Utente
Pre-conditions	L'utente deve avere un account registrato
Sequenza principale	1. L'utente fornisce le credenziali. 2. Il sistema verifica l'identità dell'utente. 3. L'utente accede al portale.
Sequenze alternative	Le credenziali fornite sono errate e l'utente non riesce ad accedere.
Post-conditions	L'utente è autenticato nel sistema.

Titolo	2.Scrivere una storia
Attori	3.Utente
Pre-conditions	L'utente deve essere autenticato.
Sequenza principale	1. L'utente inizia una nuova storia. 2. L'utente fornisce il titolo 3. La storia viene registrata nel sistema
Sequenze alternative	3.1 L'utente scrive delle scene
Post-conditions	La nuova storia è creata e salvata nel sistema

Titolo	3. Creare un oggetto
Attori	Utente
Pre-conditions	L'oggetto non deve essere già nel sistema
Sequenza principale	1. L'utente inserisce il nome dell'oggetto 2. L'oggetto viene salvato nel sistema.
Sequenze alternative	
Post-conditions	

Titolo	4. Giocare una storia
Attori	Utente
Pre-conditions	
Sequenza principale	1. L'utente sceglie una strada 2. L'utente arriva ad uno scenario finale.
Sequenze alternative	1.1 L'utente risponde ad un indovinello 1.2 L'utente raccoglie un oggetto
Post-conditions	Lo stato della partita viene salvato

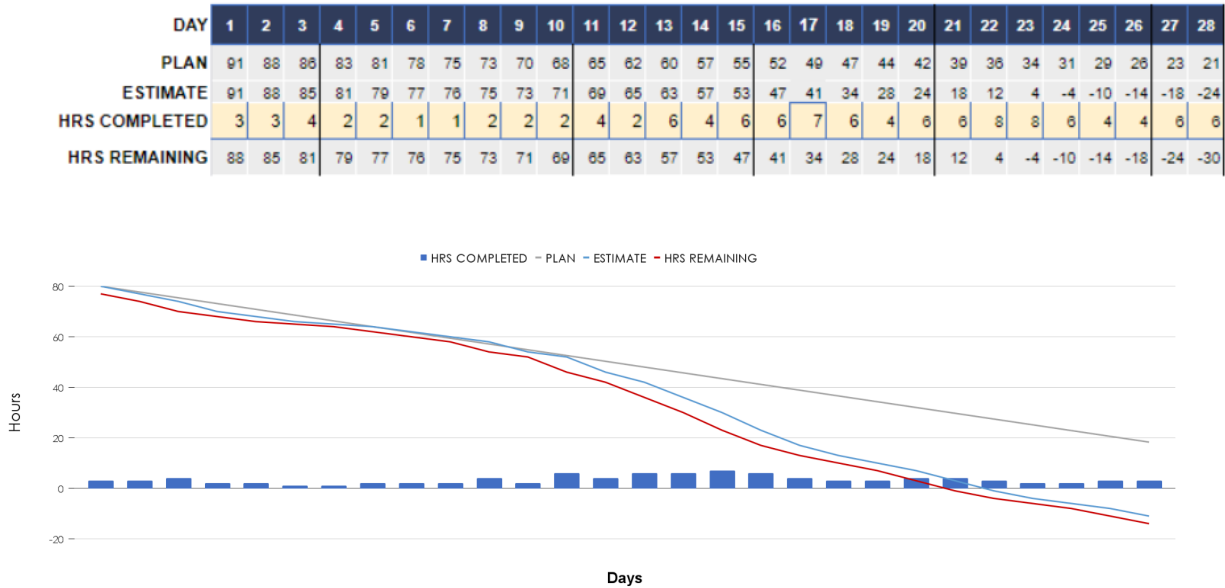
Titolo	5. Gestire storie in corso
Attori	Utente
Pre-conditions	L'utente deve aver giocato alla storia
Sequenza principale	1. L'utente sceglie una storia 2. L'utente cancella il salvataggio relativo alla storia
Sequenze alternative	
Post-conditions	Il salvataggio della partita viene cancellato

Titolo	6. Modificare il testo di uno scenario
Attori	Utente
Pre-conditions	L'utente deve aver creato almeno uno scenario.
Sequenza principale	1. L'utente sceglie una storia 2. L'utente seleziona lo scenario da modificare. 3. L'utente modifica il testo dello scenario. 4. Il nuovo testo viene salvato.
Sequenze alternative	
Post-conditions	Il testo dello scenario è aggiornato.

Titolo	7. Visualizzare elenco storie
Attori	Utente, Ospite
Pre-conditions	
Sequenza principale	1. L'utente visualizza le storie disponibili. 2. L'utente vede la prima scena della storia
Sequenze alternative	
Post-conditions	

Titolo	8. Giocare una partita
Attori	Utente
Pre-conditions	L'utente deve aver selezionato una storia da giocare.
Sequenza principale	1. L'utente legge lo scenario corrente. 2. L'utente prende una decisione e sceglie lo scenario successivo. 3. L'utente raccoglie l'oggetto se presente 4. Il sistema presenta il prossimo scenario basato sulla scelta.
Sequenze alternative	2.1 L'utente risponde all'indovinello e prosegue allo scenario successivo. 3.1 L'utente non possiede l'oggetto/i necessario/i per continuare su un certo scenario, per cui ne sceglie un altro
Post-conditions	La partita continua fino al raggiungimento di un finale.

Burndown Chart




Manuale dell'Utente

Login:

Schermata per autenticarsi ed entrare nella piattaforma.

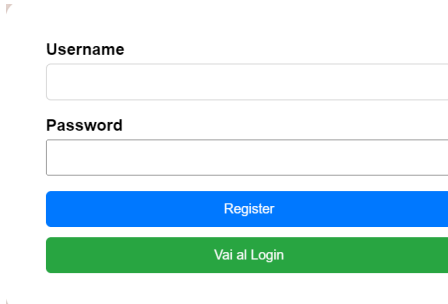
Offre la possibilità di entrare nella pagina di registrazione o entrare nella piattaforma come ospite



Register:

Schermata per creare il tuo profilo col quale accedere alla piattaforma.

Offre la possibilità di tornare alla schermata di accesso

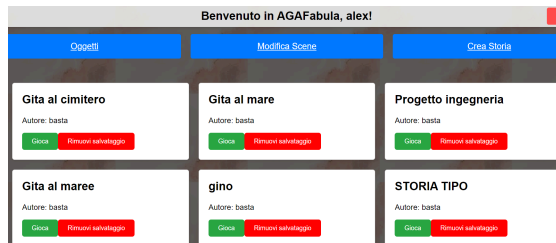


Home:

Pagina principale dell'applicazione.

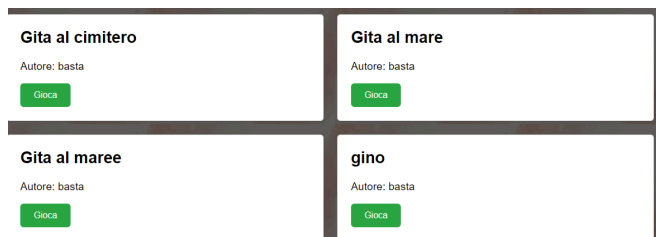
Offre la possibilità di:

- Giocare alle diverse storie
- Cancellare i salvataggi delle partite
- Accedere alla creazione oggetti
- Accedere alla modifica descrizioni
- Accedere alla creazione storia



VisitorHome:

Pagina principale per gli ospiti (Persone che non hanno eseguito l'accesso) permette di visualizzare le storie ed accedere visitorGame



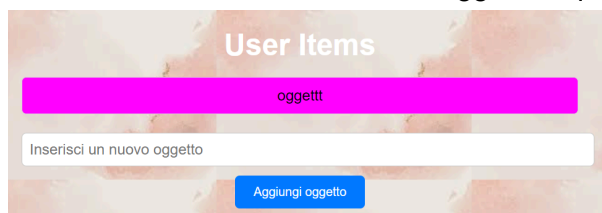
VisitorGame:

Pagina nella quale gli ospiti possono vedere la prima scena di ogni storia senza però poterci interagire



Item:

Pagina per la creazione degli oggetti.
Permette di inserire il nome di un oggetto il quale apparirà sopra la textArea

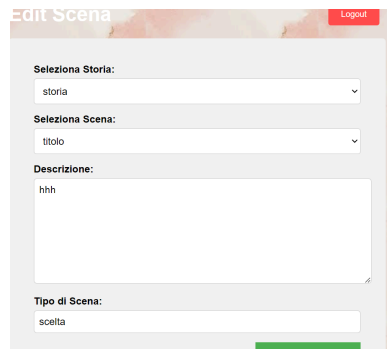


EditScene:

Pagina per la modifica della descrizione delle scene.

L'utente sceglie da un menù a tendina prima la storia di riferimento e poi la scena di riferimento facendo apparire una textarea con la relativa descrizione modificabile.

Sotto la textArea è presente anche il tipo della scena e il pulsante aggiorna descrizione



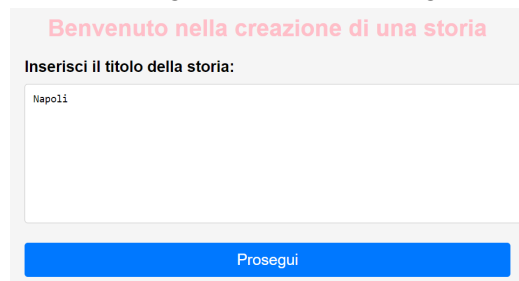
The screenshot shows a web form titled "Edit Scene" with a "Logout" button in the top right corner. The form contains three main sections: "Selezione Storia:" with a dropdown menu showing "storia"; "Selezione Scena:" with a dropdown menu showing "titolo"; and "Descrizione:" with a large text area containing the text "hhh". Below the text area is a "Tipo di Scena:" label and a dropdown menu showing "scelta". At the bottom right of the form is a green button labeled "Aggiorna Descrizione".

CreateStory:

Pagina per la creazione delle storie.

Permette l'accesso sequenziale alle pagine CreateScene, firstScene, riddle, choice.

In questa pagina è possibile scegliere il titolo della storia.



The screenshot shows a web form titled "Benvenuto nella creazione di una storia". Below the title is a label "Inserisci il titolo della storia:" followed by a text area containing the text "Napoli". At the bottom of the form is a blue button labeled "Prosegui".

CreateScene:

Pagina atta alla creazione delle scene.

E' presente una maschera contenente due textArea per Titolo scena e Descrizione; tre menù a tendina per scegliere l'oggetto trovato, l'oggetto richiesto, il tipo della scena.

Sono presenti inoltre due pulsanti Salva scena e Prosegui.

Il primo dopo aver controllato che non sia già presente una scena con lo stesso titolo salva la scena e resetta i campi.

Il secondo pulsante controlla che sia stata creata almeno una scena finale e indirizza alla pagina firstScene.

Storia: napoli

Autore: alex

Titolo:

Salerno

Descrizione:

Reggio-calabria

Tipo di scena:

Scelta

Oggetto Richiesto:

oggetti

Oggetto Trovato:

Nessuno

Salva Scena

FirstScene:

Pagina per scegliere lo scenario iniziale di una storia.

appare un menù a tendina nel quale sono presenti tutti gli scenari della storia.

Premendo il pulsante prosegui viene aggiornato l'attributo "firstScene" di storia con lo scenario scelto, per poi essere indirizzati verso riddle.

Scegli la tua scena iniziale per la storia: napoli

Autore: alex

Seleziona la scena:

Salerno

Scena selezionata: Salerno

Prosegui

Riddle

Pagina contenente i titoli di tutte le scene indovinello della storia.

Ogni scenario è contenuto in una maschera contenente una textarea per l'indovinello, una textarea per la risposta corretta e due menù a tendina.

Nel primo viene scelta la scena nella quale si arriva dando la risposta corretta, il secondo dando la risposta sbagliata.

il pulsante prosegui conduce alla pagina choice.

Indovinelli per la Storia: napoli

Indovinello

Testo dell'Indovinello:

Risposta:

Scena Corretta:

Seleziona una scena ▾

Scena Sbagliata:

Seleziona una scena ▾

marco tognò

Testo dell'Indovinello:

Risposta:

Scena Corretta:

Seleziona una scena ▾

Scena Sbagliata:

Seleziona una scena ▾

Choice:

Pagina per associare ad ogni scenario di tipo scelta le diverse destinazioni.

Per ogni scenario di tipo scelta appare una maschera contenente due menù a tendina nei quali scegliere le destinazioni.

Appare inoltre un pulsante “aggiungi destinazione” che aggiunge ulteriori destinazioni da associare alla scena

Il pulsante Salva storia salverà le associazioni e riporterà alla schermata home.

Scelte per la Storia: napoli

il ponte di trastevere

Destinazione:

Seleziona una scena ▾

Destinazione:

Seleziona una scena ▾

Aggiungi Destinazione

Prosegui

Game:

Pagina nella quale viene caricato il primo scenario e in base alla scena attuale viene aggiornato l'attributo “scenario iniziale” all'interno della tabella “Games” presente nel database.

Dentro gioca storia, in base al tipo di scenario attuale appare una diversa schermata.

Se lo scenario è di tipo scelta apparirà il titolo e la descrizione della scena e dei pulsanti che cambiano lo scenario in quello scelto.

Se lo scenario è di tipo indovinello, oltre al titolo e alla descrizione apparirà il testo dell'indovinello seguito da una textArea nella quale scrivere la risposta.

In base alla risposta si aggiorna lo scenario attuale.

In alcune scene possono apparire degli oggetti, in questo caso appare un pop-up che notifica l'utente dell'acquisizione dell'oggetto. Se si ricapita nello stesso scenario viene fatto un check

della presenza dell'oggetto nell'inventario, e, se presente, viene nascosto il pop-up e l'oggetto non viene aggiunto.

Se per accedere ad uno scenario è necessario un oggetto non posseduto appare una notifica rossa dove viene detto l'oggetto richiesto, senza aggiornare la scena.

Se si trova in uno scenario finale l'utente vede titolo della scena e descrizione.

L'utente può tornare alla schermata home in qualunque momento. il salvataggio dei dati avviene durante il cambio scena, aggiornando lo scenario.

Scena iniziale

Descrizione iniziale

Titolo indovinello Titolo scelta

Object found: ossa

Close

Titolo indovinello

Descrizione indovinello

Testo indovinello (Risposta: napoli)

Submit

Manuale dello Sviluppatore

Setup e Deploy

Esecuzione del Progetto

1. Apri un terminale o prompt dei comandi sul PC di destinazione.
 2. Naviga nella directory principale del progetto, dove si trova il file docker-compose.yml.
 - Usa il comando `cd` per cambiare directory. Ad esempio:
`cd /path/to/your/project`
 3. Assicurati che il file docker-compose.yml sia presente nella directory corrente.
 - Puoi verificare questo con il comando `ls` (Mac/Linux) o `dir` (Windows).
 4. Esegui il comando per avviare i servizi Docker:
`docker-compose up --build`
 5. Docker Compose inizierà a costruire le immagini Docker e ad avviare i container.
 - Questo potrebbe richiedere alcuni minuti a seconda della velocità di Internet e delle risorse del PC.
 6. Una volta che tutti i servizi sono avviati, apri un browser web sul PC di destinazione.
 - Per accedere al frontend, vai a `http://localhost:3000`
 - Per accedere al backend (se necessario), vai a <http://localhost:5001>
- Alternativamente, se non si è in possesso di Docker, per avviare correttamente l'applicazione è necessario seguire i seguenti passaggi

1. Configurazione del Database MongoDB:

Assicurarsi che MongoDB sia installato sulla macchina locale.

Avviare il server MongoDB con il comando: mongod

2. Avvio del Backend:

Aprire un terminale e navigare nella directory del backend:

cd /workspace/Ingsoft/backend/src

Avviare il server Node.js con il comando:

node server.js

il backend sarà accessibile di default all'indirizzo <http://localhost:5001>.

3. Avvio del Frontend:

Aprire un altro terminale e navigare nella directory del frontend:

cd /workspace/Ingsoft/frontend/src

Avviare il server di sviluppo del frontend con il comando:

npm start

L'applicazione frontend sarà accessibile di default all'indirizzo <http://localhost:3000>.

Struttura del Progetto

- **Frontend:** Sviluppato con React, il frontend è organizzato in componenti e utilizza CSS per la stilizzazione. La struttura dei file è suddivisa in pagine e componenti per una gestione modulare della UI.
- **Backend:** Sviluppato con Node.js e Express, il backend è diviso in modelli (models) e rotte (routes). La cartella principale contiene il file server.js che avvia il server e gestisce la connessione al database.
- **Database:** MongoDB è utilizzato come sistema di gestione del database, con Mongoose come ODM per la gestione dei dati.

Dettagli sui Servizi e Configurazione

Backend

Il backend è implementato seguendo l'architettura MVC:

- **Models:** Definiscono la struttura dei dati nel database MongoDB. Ogni modello rappresenta una raccolta, come User, Game, Story.
- **Routes:** Gestisce le operazioni CRUD e altre funzionalità API, separate per ogni entità.
- **Server:** Configurazione di Express nel file server.js, inclusa la gestione delle connessioni al database e l'ascolto delle richieste in arrivo.

Modern Pattern e Principi Utilizzati

Single Responsibility Principle (SRP):

Ogni componente del sistema ha una singola responsabilità, evidenziata dalla separazione delle API nelle rotte specifiche per ciascun modello nel backend.

Modularità:

Il codice è strutturato in moduli distinti, con il frontend organizzato in componenti React riutilizzabili e il backend suddiviso in modelli (models) e rotte (routes).

Model-View-Controller (MVC):

Il progetto segue il pattern MVC, suddividendo la logica del progetto in:

Model: gestione della logica di business e delle interazioni con MongoDB tramite Mongoose.

View: gestita dal frontend React, che cura la presentazione e l'interazione utente.

Controller: implementato nelle routes del backend, che controllano le richieste HTTP e inviano risposte appropriate.

Separazione dei Concerns:

È stato seguito il principio della separazione dei concerns, isolando le diverse funzionalità del sistema in sezioni distinte (ad es., frontend e backend, modelli e rotte).

Client-Server Architecture:

Il progetto utilizza un'architettura client-server, con il frontend che agisce da client e il backend come server, comunicando tramite richieste HTTP RESTful.

RESTful API Design:

Le API del backend sono progettate seguendo i principi RESTful, utilizzando i metodi HTTP per operazioni CRUD e offrendo un'interfaccia consistente e standardizzata.

Component-Based Architecture (nel frontend con React):

L'approccio a componenti in React permette di suddividere l'interfaccia utente in parti indipendenti e riutilizzabili, migliorando la modularità e la gestione dello stato.

Asynchronous Programming:

La programmazione asincrona è utilizzata per gestire operazioni I/O non bloccanti, come le interazioni con il database e le chiamate API, migliorando l'efficienza e la reattività dell'applicazione.

Diario del progetto

Sprint 0

Descrizione:

Nella fase iniziale dello sviluppo ci siamo trovati e abbiamo fatto la dichiarazione d'intenti, leggendo insieme la traccia e condividendo idee sulla realizzazione.

Lo sprint 0 è stato utile anche a suddividere gli sprint successivi, avere un setup uguale per tutti all'interno dei diversi computer e sviluppare il diagramma dei casi d'uso e il diagramma di dominio.

Sprint 1

Periodo: 01/07/2024 - 05/07/2024

Ruoli:

- Software Developer: Alex Ranaulo, Alessandro Desolda
- Product Owner: Gioele Blanc
- Scrum Master: Gioele Blanc

Descrizione: Nel primo sprint ci siamo concentrati sulla fase di SignIn e LogIn, sviluppando contemporaneamente FrontEnd e BackEnd

FASE INIZIALE

UC Raffinati	Product Backlog	Sprint Backlog
	Login/SignIn (UC1)	Creazione models Utente(UC1)
	Creare Oggetti (UC2)	Creazione routes Utente(UC1)
	Creazione Scene (UC3)	Creazione pagina registrazione (UC1)
	Aggiornare Decrizione (UC4)	Creazione pagina Login (UC1)
	Scrivere Storia (UC5)	Creazione pagina SignIn (UC1)
	Visualizzare elenco storie (UC6)	
	Giocare storia (UC7)	
	Cancellare storie (UC8)	

To do	In progress	Done
Creazione models Utente(UC1)		
Creazione routes Utente(UC1)		
Creazione pagina registrazione (UC1)		
Creazione pagina Login (UC1)		
Creazione pagina SignIn (UC1)		

FASE FINALE

UC Raffinati	Product Backlog	Sprint Backlog
Login/SignIn (UC1)	Creare Oggetti (UC2)	
	Creazione Scene (UC3)	
	Aggiornare Decrizione (UC4)	
	Scrivere Storia (UC5)	
	Visualizzare elenco storie (UC6)	
	Giocare storia (UC7)	
	Cancellare storie (UC8)	

To do	In progress	Done
		Creazione models Utente(UC1)
		Creazione routes Utente(UC1)
		Creazione pagina registrazione (UC1)
		Creazione pagina Login (UC1)
		Creazione pagina SignIn (UC1)

Sprint 2

Periodo: 08/07/2024 - 12/07/2024

Ruoli:

- Software Developer: Alex Ranaulo, Alessandro Desolda
- Product Owner: Gioele Blanc
- Scrum Master: Gioele Blanc

Descrizione: Nel secondo sprint abbiamo lavorato sulla creazione degli oggetti e delle scene, nonché al loro salvataggio nel DataBase

FASE INIZIALE

UC Raffinati	Product Backlog	Sprint Backlog
Login/SignIn (UC1)	Creare Oggetti (UC2)	Creazione model Oggetto (UC2)
	Creazione Scene (UC3)	Creazione routes oggetto (UC2)
	Aggiornare Decrizione (UC4)	Creazione BE Oggetto (UC2)
	Scrivere Storia (UC5)	Creazione FE Oggetto (UC2)
	Visualizzare elenco storie (UC6)	Creazione model Scena (UC2)
	Giocare storia (UC7)	Creazione routes Scena (UC2)
	Cancellare storie (UC8)	Creazione BE Scena (UC2)
		Creazione FE Scena (UC2)

To do	In progress	Done
Creazione model Oggetto (UC2)		Creazione models Utente(UC1)
Creazione routes oggetto (UC2)		Creazione routes Utente(UC1)
Creazione BE Oggetto (UC2)		Creazione pagina registrazione (UC1)
Creazione FE Oggetto (UC2)		Creazione pagina Login (UC1)
Creazione model Scena (UC2)		Creazione pagina SignIn (UC1)
Creazione routes Scena (UC2)		
Creazione BE Scena (UC2)		
Creazione FE Scena (UC2)		

FASE FINALE

UC Raffinati	Product Backlog	Sprint Backlog
Login/SignIn (UC1)	Aggiornare Decrizione (UC4)	Creazione model Oggetto (UC2)
Creare Oggetti (UC2)	Scrivere Storia (UC5)	Creazione routes oggetto (UC2)
Creazione Scene (UC3)	Visualizzare elenco storie (UC6)	Creazione BE Oggetto (UC2)
	Giocare storia (UC7)	Creazione FE Oggetto (UC2)
	Cancellare storie (UC8)	Creazione model Scena (UC2)
		Creazione routes Scena (UC2)
		Creazione BE Scena (UC2)
		Creazione FE Scena (UC2)

To do	In progress	Done
		Creazione models Utente(UC1)
		Creazione routes Utente(UC1)
		Creazione pagina registrazione (UC1)
		Creazione pagina Login (UC1)
		Creazione pagina SignIn (UC1)
		Creazione model Oggetto (UC2)
		Creazione routes oggetto (UC2)
		Creazione BE Oggetto (UC2)
		Creazione FE Oggetto (UC2)
		Creazione model Scena (UC2)
		Creazione routes Scena (UC2)
		Creazione BE Scena (UC2)
		Creazione FE Scena (UC2)

Sprint 3

Periodo: 15/07/2024 - 20/07/2024

Ruoli:

- Software Developer: Alessandro Desolda, Gioele Blanc
- Product Owner: Alex Ranaulo
- Scrum Master: Alex Ranaulo

Descrizione: Nel terzo sprint ci siamo concentrati sulla creazione della storia e sulla modellazione dei salvataggi, nonché della creazione pagina ospite.

FASE INIZIALE

UC Raffinati	Product Backlog	Sprint Backlog
Login/SignIn (UC1)	Aggiornare Decrizione (UC4)	Creazione FE descrizione (UC4)
Creare Oggetti (UC2)	Scrivere Storia (UC5)	Creazione BE choice (UC5)
Creazione Scene (UC3)	Visualizzare elenco storie (UC6)	Creazione BE riddle (UC5)
	Giocare storia (UC7)	Creazione BE story (UC5)
	Cancellare storie (UC8)	Creazione FE choice (UC5)
		Creazione FE riddle (UC5)
		Creazione FE story (UC5)
		module e route story (UC5)
		module e route riddle (UC5)
		module e route choice (UC5)
		Visualizza elenco storie (UC6)
		Creazione BE game (UC7)
		Creazione FE game (UC7)
		module e route game (UC7/UC8)

To do	In progress	Done
Creazione FE descrizione (UC4)		Creazione models Utente(UC1)
Creazione BE choice (UC5)		Creazione routes Utente(UC1)
Creazione BE riddle (UC5)		Creazione pagina registrazione (UC1)
Creazione BE story (UC5)		Creazione pagina Login (UC1)
Creazione FE choice (UC5)		Creazione pagina Signin (UC1)
Creazione FE riddle (UC5)		Creazione model Oggetto (UC2)
Creazione FE story (UC5)		Creazione routes oggetto (UC2)
module e route story (UC5)		Creazione BE Oggetto (UC2)
module e route riddle (UC5)		Creazione FE Oggetto (UC2)
module e route choice (UC5)		Creazione model Scena (UC2)
Visualizza elenco storie (UC6)		Creazione routes Scena (UC2)
Creazione BE game (UC7)		Creazione BE Scena (UC2)
Creazione FE game (UC7)		Creazione FE Scena (UC2)
module e route game (UC7/UC8)		

FASE FINALE

To do	In progress	Done
		Creazione models Utente(UC1)
		Creazione routes Utente(UC1)
		Creazione pagina registrazione (UC1)
		Creazione pagina Login (UC1)
		Creazione pagina SignIn (UC1)
		Creazione model Oggetto (UC2)
		Creazione routes oggetto (UC2)
		Creazione BE Oggetto (UC2)
		Creazione FE Oggetto (UC2)
		Creazione model Scena (UC2)
		Creazione routes Scena (UC2)
		Creazione BE Scena (UC2)
		Creazione FE Scena (UC2)
		Creazione FE descrizione (UC4)
		Creazione BE choice (UC5)
		Creazione BE riddle (UC5)
		Creazione BE story (UC5)
		Creazione FE choice (UC5)
		Creazione FE riddle (UC5)
		Creazione FE story (UC5)
		module e route story (UC5)
		module e route riddle (UC5)
		module e route choice (UC5)
		Visualizza elenco storie (UC6)
		Creazione BE game (UC7)
		Creazione FE game (UC7)
		module e route game (UC7/UC8)

Periodo: 22/07/2024 - 23/07/2024

- Software Developer: Alex Ranaulo, Gioele Bianco
- Product Owner: Alessandro Desolda,
- Scrum Master: Alessandro Desolda,

[illegible][illegible]