

SDD

System Design Document



BeYourChoice

Riferimento	BYC_RAD_V_1.5
Versione	1.2
Data	16/11/2024
Destinatario	Prof.ssa Filomena Ferrucci, Prof. Fabio Palomba
Presentato da	Tutti i team member
Approvato da	Marco Ciano, Giuseppe D'Avino

Revision History

Data	Versione	Descrizione	Autori
12/11/2024	1.0	Aggiunta dell'introduzione e dei design goal	Marco Acierno Iari Normanno Federica Iuliano Rocco Cione Augusto Persico
12/11/2024	1.0	Aggiunta sistema corrente: panoramica; Sistema proposto: panoramica, decomposizione in sottosistemi e decomposizione in layer	Marco Acierno Iari Normanno Aldo Damiano Antonio Rapa Augusto Persico Rocco Cione
12/11/2024	1.0	Aggiunto mapping hardware/software	Fabrizio Cozzolino
13/11/2024	1.0	Correzione paragrafo 3.1; aggiunta dei trade-off; gestione dei dati persistenti	Marco Acierno Iari Normanno Federica Iuliano Rocco Cione Aldo Damiano Fabrizio Cozzolino Augusto Persico Antonio Rapa
13/11/2024	1.1	Aggiunto controllo flusso globale; Condizioni Boundary	Marco Acierno Iari Normanno Aldo Damiano Antonio Rapa



15/11/2024	1.1	Aggiunto il diagramma del servizio di sottosistemi con un elenco dei servizi; completamento della gestione dei dati persistenti e delle condizioni boundary; revisione trade off e design goal	Marco Acierno Iari Normanno Federica Iuliano Rocco Cione Aldo Damiano Fabrizio Cozzolino Augusto Persico Antonio Rapa
16/11/2024	1.2	Revisione completa del documento	Marco Acierno Iari Normanno Federica Iuliano Rocco Cione Aldo Damiano Fabrizio Cozzolino Augusto Persico Antonio Rapa

Sommario

1 Introduzione	6
1.1 Obiettivi del Sistema	6
1.2 Design Goals	7
1.3 Definizioni, Acronimi e Abbreviazioni	12
1.4 Riferimenti	12
1.5 Overview	13
2 Architettura del Sistema Corrente	13
2.1 Panoramica	13
3 Architettura del Sistema Proposto	15
3.1 Panoramica	15
3.2 Decomposizione in sottosistemi	16
3.2.1 Decomposizione in layer	16
3.2.2 Decomposizione in sottosistemi	18
3.3 Mapping Hardware/Software	19
3.4 Gestione dei dati persistenti	20
3.4.1 Descrizione delle entità persistenti	20
3.4.1.1 Studente	20
3.4.1.2 Docente	21
3.4.1.3 Dashboard	22
3.4.1.4 Classe Virtuale	22
3.4.1.5 Materiale Didattico	23
3.4.1.6 Scenario Virtuale	23
3.4.1.7 Punteggio Scenario	24
3.4.1.8 Quiz	24
3.4.1.9 Domanda	25
3.4.1.10 Risultato Quiz	25
3.5 Controllo della sicurezza	26
3.6 Controllo flusso globale del sistema BB	28
3.7 Condizioni Boundary	29
3.7.1 Avvio del sistema	29
3.7.2 Arresto del sistema	30

3.7.3 Fallimento del sistema	32
4 Servizi di sottosistemi	33
4.1 GUI Autenticazione - 3 Servizi per Control	33
4.2 GUI Profilo - 3 Servizi per Control	34
4.3 GUI Quiz - 4 Servizi per Control	34
4.4 GUI Classe - 3 Servizi per Control	34
4.5 GUI Scenario Virtuale - 1 Servizi per Control	34
4.6 GUI Dashboard - 3 Servizi per Control	34
4.7 GUI Materiale Didattico - 4 Servizi per Control	34
4.8 Gestione Classe - 3 Servizi per Model	35
4.9 Gestione Scenario Virtuale - 1 Servizio per Model	35
4.10 Gestione Autenticazione - 2 Servizi per Model	35
4.11 Gestione Profilo - 3 Servizi per Model	35
4.12 Gestione Quiz - 4 Servizi per Model	35
4.13 Gestione Classe - 3 Servizi per Model	36
4.14 Gestione Scenario Virtuale - 1 Servizi per Model	36
4.15 Gestione Dashboard - 3 Servizi per Model	36
4.16 Gestione Materiale Didattico - 4 Servizi per Model	36

1 Introduzione

1.1 Obiettivi del Sistema

Il sistema BeYourChoice ha come obiettivo la creazione di una piattaforma web interattiva dedicata all'insegnamento dell'educazione civica per gli studenti delle scuole superiori. Si punta a rendere l'apprendimento più coinvolgente e interattivo utilizzando tecniche di gamification e simulazioni politiche. La gamification prevede l'uso di meccaniche di gioco per aumentare la motivazione degli studenti, rendendo le attività educative più divertenti e stimolanti. Le simulazioni politiche permettono agli studenti di partecipare attivamente a scenari virtuali che riproducono situazioni legate ai processi democratici, come elezioni, dibattiti parlamentari o decisioni legislative. Questi strumenti hanno lo scopo di sviluppare il pensiero critico, migliorare la capacità di cooperazione tra studenti e far comprendere meglio il funzionamento delle istituzioni politiche e dei processi democratici.

La piattaforma mette a disposizione funzionalità per permettere ai docenti di configurare quiz e attività interattive, fornendo un feedback immediato sui progressi degli studenti. I quiz possono essere personalizzati in base agli argomenti trattati e il sistema include una componente che genera automaticamente domande pertinenti a quest'ultimi, così da facilitare ulteriormente il lavoro dei docenti. Gli studenti possono accedere a questi quiz per valutare le proprie conoscenze e ricevere il punteggio sulle risposte date. Inoltre, il sistema consente di caricare e visualizzare materiale didattico, come documenti, video e presentazioni, che gli studenti possono consultare per approfondire le tematiche studiate.

In conclusione, il sistema BeYourChoice mira a creare un ambiente di apprendimento dinamico e partecipativo, dove studenti e docenti possono interagire attivamente, sfruttando la tecnologia, per approfondire la comprensione dei principi democratici e delle istituzioni politiche. L'approccio basato su gamification e simulazioni politiche non solo rende l'apprendimento più accattivante, ma incoraggia anche una partecipazione attiva e un maggiore impegno nello studio dell'educazione civica.

1.2 Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine	Trade-off
1	DG_1 Response Time	Il sistema si deve prestare a soddisfare le richieste degli utenti in meno di 20 secondi, trasmettendo all'utente sensazione di fluidità e immediatezza per i dati richiesti.	Performance	RNF_3	Response Time vs. Security Il sistema darà priorità alla sicurezza rispetto al tempo di risposta, implementando controlli aggiuntivi per proteggere i dati degli utenti, anche se ciò potrà rallentare leggermente l'esperienza.
2	DG_2 Robustness	Il sistema deve garantire robustezza, gestendo nel modo previsto il 98% degli input errati da parte dell'utente.	Dependability	RNF_2	Robustness vs. Usability La robustezza sarà prioritaria rispetto all'usabilità, implementando controlli stringenti per validare gli input degli utenti e prevenire errori o comportamenti imprevisti, anche a scapito di una maggiore complessità dell'interfaccia.

3	DG_3 Reliability	Il sistema deve garantire l'affidabilità dei servizi proposti.	Dependability	RNF_2	Reliability vs. Changeability L'affidabilità sarà preferita alla modificabilità, adottando un approccio che limita i cambiamenti frequenti per garantire stabilità e continuità operativa, riducendo il rischio di regressioni o comportamenti inattesi.
4	DG_4 Security	Il sistema deve fornire una connessione sicura e cifrata, al fine di salvaguardare tutti i dati sensibili degli utenti da accessi non autorizzati.	Dependability	RNF_2	
5	DG_5 Availability	Il sistema deve essere sempre accessibile, salvo durante le manutenzioni programmate dagli amministratori di sistema.	Dependability	RNF_2	Availability vs. Portability Il sistema darà priorità alla disponibilità, assicurando che l'applicazione web sia sempre accessibile tramite browser sui PC degli utenti. Tuttavia, questa scelta riduce

					la portabilità, poiché il sistema non sarà ottimizzato per l'uso su dispositivi diversi dai PC, come smartphone o tablet.
6	DG_6 Fault Tolerance	Il sistema permette il salvataggio dei dati in modo automatico, così da evitare problemi in caso di fallimento.	Dependability	RNF_2	
7	DG_7 Extensibility	Assicurare che il sistema sia facilmente scalabile per aggiungere nuove funzionalità.	Maintenance	RNF_4	Extensibility vs. Response Time Il sistema privilegerà l'estendibilità rispetto al tempo di risposta, adottando una progettazione modulare e scalabile che facilita l'aggiunta di nuove funzionalità, anche se ciò può comportare un lieve impatto sulle prestazioni in tempo reale.

8	DG_8 Changeability	Il sistema non permette agli utenti di modificare le sue funzionalità.	Maintenance	RNF_4	
9	DG_9 Portability	L'interazione con il sistema avviene tramite browser poiché è sviluppata come una web application. È accessibile esclusivamente dai pc in possesso di un browser qualsiasi.	Maintenance	RNF_7	
10	DG_10 Requirements Traceability	I requisiti sono tracciabili attraverso una matrice di tracciabilità.	Maintenance	RNF_4	
11	DG_11 Usability	Il sistema prevede un'interfaccia ordinata, user-friendly e di facile utilizzo.	End User	RNF_1	

12	DG_12 Utility	Il sistema contribuisce a migliorare l'apprendimento e la pratica dell'educazione civica all'interno degli istituti scolastici, rendendola più efficace e coinvolgente.	End User	RNF_1	
----	------------------	---	----------	-------	--

1.3 Definizioni, Acronimi e Abbreviazioni

Acronimi:

- **RAD** = Requirement Analysis Document;
- **SDD** = System Design Document;
- **DG** = Design Goal;
- **RNF** = Requisito Non Funzionale;
- **GUI** = Graphical User Interface;
- **DBMS** = Database Management System;
- **MVC** = Model View Controller;
- **I.U** = Identificativo Univoco;
- **UC** = Use Case;
- **UCB** = Use Case Boundary;
- **C.F** = Codice Fiscale;
- **SdA** = Scuola di appartenenza.

1.4 Riferimenti

Libro = *Object Oriented Software Engineering (Using UML, Patterns and Java, Prentice Hall)*

Autori = *B. Bruegge & A.H. Dutoit.*

Requirements Analysis Document (BYC_RAD_V_1.5).

Slide del corso, presenti sulla piattaforma e-learning.

1.5 Overview

Il seguente documento è diviso in sezioni e ha la seguente composizione:

- **Introduzione:** In questa sezione viene descritto in generale lo scopo del sistema e gli obiettivi di design che il sistema propone di raggiungere.
- **Architettura del Sistema Corrente:** In questa sezione viene descritto lo stato attuale dell'architettura del sistema corrente già presente.
- **Architettura del Sistema Proposto:** Viene presentata l'architettura del sistema proposto, nella quale sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema e le condizioni limite.
- **Servizi dei Sottosistemi:** Vengono presentati i servizi offerti dai sottosistemi.

2 Architettura del Sistema Corrente

2.1 Panoramica

Attualmente, non esiste una piattaforma digitale dedicata all'insegnamento dell'educazione civica nelle scuole italiane che integri tecnologie interattive o elementi di gamification per coinvolgere gli studenti attivamente. L'approccio educativo per questa materia si basa su metodi tradizionali e, in aula, l'insegnante è il principale mediatore del contenuto. Le lezioni si svolgono generalmente in modalità frontale, con il docente che espone i concetti e modera le discussioni, mentre gli studenti partecipano come ascoltatori, prendendo appunti, esponendo eventuali domande e studiando il materiale a casa.

Il materiale didattico fornito ai ragazzi include testi di studio, come libri di educazione civica, articoli di approfondimento e dispense create dai docenti. Questi materiali sono disponibili principalmente in formato fisico o, occasionalmente, in digitale, tramite file PDF o documenti Word. Tuttavia, la distribuzione di risorse digitali è gestita in modo frammentario tramite email o piattaforme generiche di condivisione file, come Google Drive, senza un sistema centralizzato che consenta al docente di monitorare facilmente l'accesso e l'effettiva fruizione dei contenuti.

La valutazione delle conoscenze acquisite dagli studenti avviene attraverso verifiche scritte, interrogazioni orali o test strutturati, che solitamente si svolgono periodicamente, a distanza di giorni o settimane dall'introduzione dei concetti in aula. Questo metodo di valutazione, distanziato nel tempo rispetto al momento dell'apprendimento, non consente un feedback immediato e mirato sui progressi dello studente e rende più difficile per il docente individuare in tempo reale le difficoltà di ciascuno studente.

Il sistema corrente non dispone di una piattaforma basata sulla gamification che renda l'apprendimento dell'educazione civica più coinvolgente e interattivo. Mancano meccanismi per incentivare la partecipazione attiva degli studenti, ad esempio tramite sfide, assegnazione di punti, giochi o simulazioni di scenari politici in cui possano assumere ruoli e prendere decisioni in contesti controllati. Queste carenze riducono le opportunità di applicare concretamente i principi civici e di sviluppare competenze come il pensiero critico e la riflessione civica.

In conclusione, l'architettura del sistema educativo corrente è limitata dall'assenza di una piattaforma integrata e interattiva. Questa carenza evidenzia la necessità di un sistema più avanzato, come quello proposto da BeYourChoice, che possa centralizzare i materiali

didattici, valutare tempestivamente le competenze degli studenti monitorando il loro progresso personale, e utilizzare elementi di gamification e simulazione per incentivare un apprendimento più attivo e immersivo.

3 Architettura del Sistema Proposto

3.1 Panoramica

Il sistema proposto consiste in un'applicazione web, accessibile dagli studenti e dai docenti, che mira a facilitare l'insegnamento e l'apprendimento dell'educazione civica negli istituti di scuola superiore in Italia. Gli utenti della nostra applicazione saranno: Studente e Docente.

Tutti gli utenti dovranno poter registrarsi e autenticarsi alla piattaforma tramite credenziali personali. Una volta effettuato il login saranno disponibili diverse funzionalità a secondo del tipo di utente. Il docente potrà gestire classi virtuali che avranno come funzionalità: la creazione e l'inserimento manuale degli studenti, oltre alla possibilità di caricare materiale didattico. I docenti potranno inoltre gestire il materiale didattico, e avranno a disposizione strumenti per creare quiz personalizzati e scenari interattivi. Questi scenari permettono agli studenti di assumere ruoli specifici e di partecipare a simulazioni di eventi. Gli studenti, a loro volta, potranno accedere al materiale didattico caricato dai docenti, partecipare attivamente a quiz e scenari simulativi, e monitorare i propri progressi tramite una dashboard dedicata. Questa dashboard offre agli studenti un quadro complessivo delle loro attività e dei risultati ottenuti mentre, per il docente offre un quadro generale dell'andamento di ogni sua singola classe. Gli utenti della piattaforma potranno inoltre comunicare attraverso una chat integrata, per scambiarsi messaggi in tempo reale.

Il sistema proposto si basa sull'architettura MVC (Model-View-Controller), implementata utilizzando Python. Questa scelta risulta particolarmente adatta alle nostre necessità, poiché la separazione delle componenti tra modello, vista e controllore offre numerosi vantaggi, tra cui:

- **Modularità:** suddivide il codice in blocchi indipendenti, facilitando la gestione e l'organizzazione del progetto.
- **Manutenibilità:** il codice è facilmente aggiornabile e gestibile, consentendo modifiche rapide a ciascuna componente senza influenzare le altre.

- **Testabilità:** separando la logica di business (Model) dalla logica di presentazione (View), è possibile testare singolarmente ogni componente, migliorando l'affidabilità del sistema.
- **Estendibilità:** l'architettura facilita l'aggiunta di nuove funzionalità senza compromettere il funzionamento delle componenti esistenti.

Per la logica di business e gestione dei dati sarà utilizzato **Python**, con il supporto del framework **Flask** per la gestione del controllore e delle interazioni utente.

Per la gestione del database sarà utilizzato **MongoDB**, un database NoSQL flessibile e scalabile, perfetto per gestire dati non strutturati e facilmente integrabile con Python.

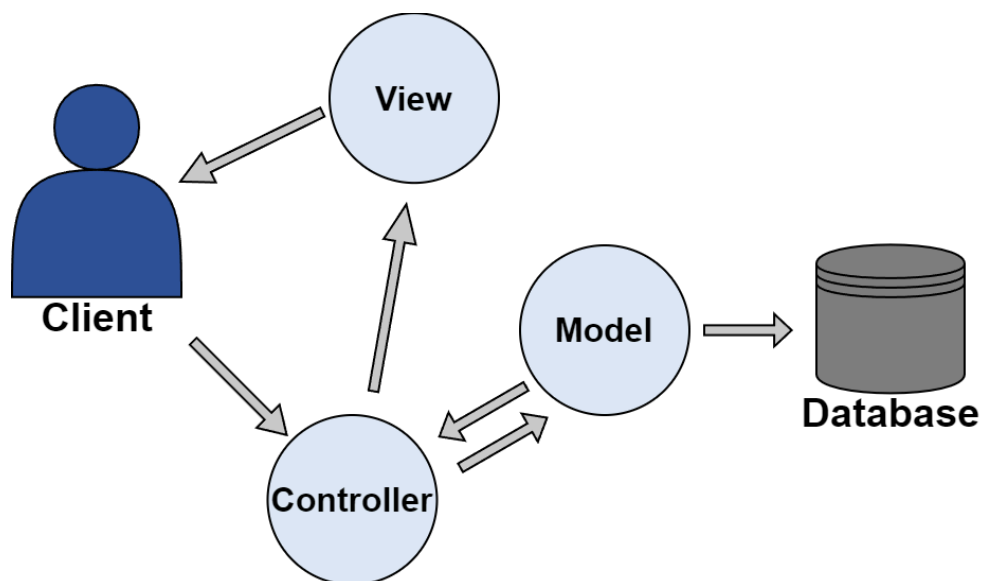
Per la logica di presentazione verranno utilizzati **HTML** e **CSS**. L'HTML sarà impiegato per strutturare le pagine e i contenuti, il CSS garantirà uno stile visivo chiaro e piacevole.

3.2 Decomposizione in sottosistemi

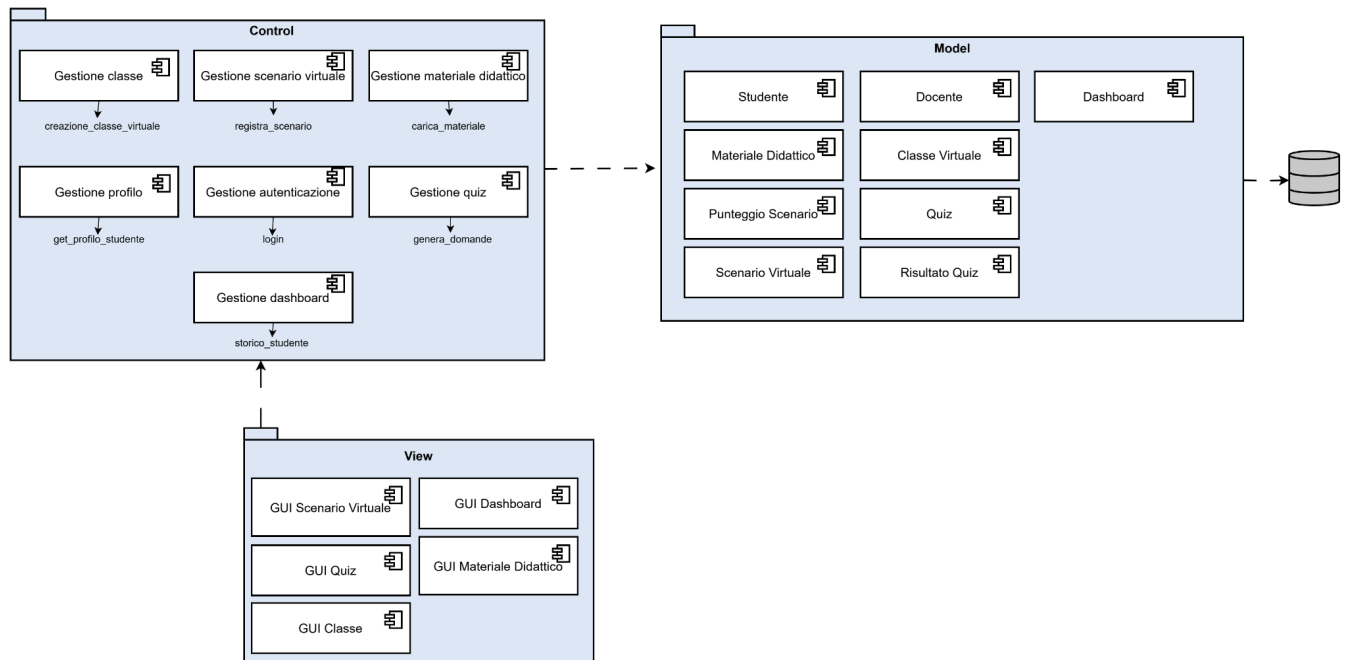
3.2.1 Decomposizione in layer

Il sistema prevede una decomposizione in tre layer. Quest'ultimi gestiscono diversi aspetti e funzionalità del sistema:

- **Model:** Questo livello rappresenta i dati e incarna la logica di business dell'applicazione. Ha il compito di gestire lo stato dell'applicazione e di rispondere alle richieste di accesso e manipolazione dei dati.
- **Controller:** Questo livello funge da "coordinatore" che gestisce l'interazione tra il Model e la View. Riceve gli input dall'utente, li elabora e determina come intervenire sul Model o aggiornare la View di conseguenza.
- **View:** Questo livello si occupa della presentazione delle informazioni all'utente. Essa accede ai dati contenuti nel Model per visualizzarli, senza tuttavia modificarli direttamente.



3.2.2 Decomposizione in sottosistemi



Il livello **View** comprende la gestione di dieci sottosistemi, definiti come oggetti di tipo "boundary" nel documento RAD, quali:

- GUI Dashboard
- GUI Quiz
- GUI Scenario Virtuale
- GUI Classe
- GUI Materiale Didattico

Il livello **Control** comprende la gestione di dieci sottosistemi, quali:

- Gestione Profilo
- Gestione Autenticazione
- Gestione Quiz
- Gestione Scenario Virtuale
- Gestione Classi
- Gestione Materiale Didattico
- Gestione Dashboard

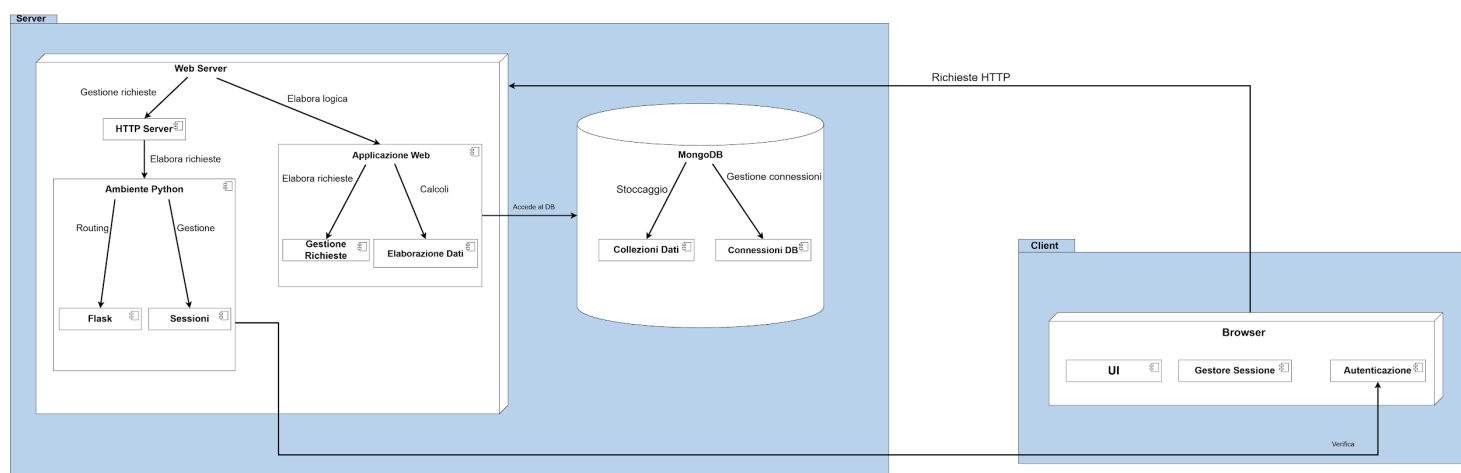
Il livello **Model** comprende la gestione di un sottosistema, quale:

- Studente
- Docente
- Dashboard
- Materiale Didattico
- Classe Virtuale
- Punteggio Scenario
- Quiz
- Scenario Virtuale
- Risultato Quiz

3.3 Mapping Hardware/Software

Il sistema adotta un'architettura Client-Server per fornire i servizi. Il Client è rappresentato dal browser, che comunica con il Server tramite richieste HTTP. Sul lato Server, è utilizzato un ambiente runtime Python per eseguire la logica applicativa e generare pagine web dinamiche che vengono inviate al browser dell'utente. Python è stato scelto per la sua versatilità e la vasta gamma di librerie e strumenti che facilitano lo sviluppo web e garantiscono scalabilità e prestazioni efficienti.

Per la gestione dei dati, il sistema utilizza MongoDB, un database NoSQL ottimizzato per l'archiviazione in formato JSON, facilitando così la gestione dei dati strutturati e non. Lo sviluppo è supportato da PyCharm, l'ambiente di sviluppo integrato scelto per la sua efficienza nel debugging e nella gestione dei progetti Python.



3.4 Gestione dei dati persistenti

Per la memorizzazione dei dati è stato scelto un database non relazionale, adottando il DBMS orientato ai documenti “MongoDB”. Questa scelta si basa su diversi vantaggi specifici di un database documentale.

In MongoDB, i dati vengono memorizzati sotto forma di documenti JSON, che permettono di conservare molte informazioni in modo flessibile. I documenti JSON possono contenere dati complessi e annidati, rendendo possibile una rappresentazione ricca e articolata delle informazioni. Questa struttura risulta particolarmente vantaggiosa per applicazioni come questa, dove i dati possono variare in base a contesti e specifiche funzioni.

I documenti sono organizzati in collezioni, che in MongoDB non richiedono uno schema fisso. Ogni documento all'interno di una collezione può avere una struttura propria, diversa dagli altri. Questo approccio offre un'elevata flessibilità, poiché le entità possono essere modificate, aggiungendo o rimuovendo attributi senza bisogno di ridefinire uno schema a livello globale, consentendo aggiornamenti dinamici e adattamenti in corso d'opera.

3.4.1 Descrizione delle entità persistenti

3.4.1.1 Studente

Entità	Studente
Descrizione	La collection 'Studente' contiene le informazioni anagrafiche dello studente, la sua e-mail, la password, il codice fiscale e la scuola di appartenenza.
I.U	C.F
Dati	
Attributi	Tipo
Nome	String
Cognome	String
SdA	String

Email	String
C.F	String
Data_Nascita	Date
Password	String
ID_Classe	Integer

3.4.1.2 Docente

Entità	Docente
Descrizione	La collection 'Docente' contiene le informazioni anagrafiche del docente, la sua e-mail, la password, il codice fiscale e la scuola di appartenenza.
I.U	C.F
Dati	
Attributi	Tipo
Nome	String
Cognome	String
SdA	String
Email	String
C.F	String
Data_Nascita	Date
Password	String
Codice_Univoco	Integer

3.4.1.3 Dashboard

Entità	Dashboard	
Descrizione	La collection 'Dashboard' contiene tutte le attività e i relativi punteggi del singolo studente.	
I.U	N/A	
Dati		
Attributi		Tipo
ID_Actività		Integer
Data_Actività		Date
Descrizione_Actività		String
Punteggio_Actività		Float

3.4.1.4 Classe Virtuale

Entità	Classe Virtuale
Descrizione	La collection 'Classe Virtuale' contiene le informazioni relative ad una classe virtuale e il suo nome.
I.U	ID_Classe
Dati	
Attributi	Tipo
ID_Classe	Integer
Nome_Classe	String
Descrizione	String
ID_Docente	String

3.4.1.5 Materiale Didattico

Entità	Materiale Didattico
Descrizione	La collection 'Materiale Didattico' contiene le informazioni relative al materiale didattico, il tipo, il titolo e una descrizione.
I.U	ID_MaterialeDidattico
Dati	
Attributi	Tipo
ID_MaterialeDidattico	Integer
Tipo	Enum
Titolo	String
Descrizione	String
ID_Classe	Integer

3.4.1.6 Scenario Virtuale

Entità	Scenario Virtuale
Descrizione	La collection 'Scenario Virtuale' contiene le informazioni relative allo scenario virtuale, il titolo, la descrizione e la modalità.
I.U	ID_Scenario
Dati	
Attributi	Tipo
ID_Scenario	Integer
Titolo	String
Descrizione	String
Argomento	String
Modalità	String

ID_Classe	Integer
-----------	---------

3.4.1.7 Punteggio Scenario

Entità	Punteggio Scenario	
Descrizione	La collection 'Punteggio Scenario' rappresenta il punteggio ottenuto da uno studente in uno scenario virtuale.	
I.U	CF_Studente ID_Scenario	
Dati		
Attributi		Tipo
CF_Studente		String
ID_Scenario		Integer
Punteggio_Scenario		Float

3.4.1.8 Quiz

Entità	Quiz
Descrizione	La collection 'Quiz' rappresenta un quiz specifico all'interno del sistema. È composto dall'argomento, numero domande, modalità e una lista di domande.
I.U	ID_Quiz
Dati	
Attributi	Tipo
ID_Quiz	Integer
Argomento	String
N_Domande	Integer

Modalità_Quiz	Enum
Domande	List<Domanda>
Durata	Float
Data_Creazione	Date
ID_Classe	Integer

3.4.1.9 Domanda

Entità	Domanda
Descrizione	La collection 'Domanda' rappresenta la singola domanda presente in un quiz con le sue risposte e la sua risposta corretta.
I.U	ID_Domanda
Dati	
Attributi	Tipo
ID_Domanda	Integer
Testo_Domanda	String
Opzioni_Risposte	Array(string)
Risposta_Corretta	String
ID_Quiz	Integer

3.4.1.10 Risultato Quiz

Entità	Risultato Quiz	
Descrizione	La collection 'Risultato Quiz' contiene le informazioni inerenti al risultato di un quiz da parte di uno studente ed è composto dal punteggio ottenuto e dall'insieme delle risposte .	
I.U	ID_Quiz	

CF_Studente	
Dati	
Attributi	Tipo
ID_Quiz	Integer
CF_Studente	String
Punteggio_Quiz	Float
Risposte	Array(String)

3.5 Controllo della sicurezza

Nel sistema "BeYourChoice", ogni attore ha il permesso di eseguire specifiche operazioni, consentendo una gestione sicura degli accessi alle varie funzionalità del software. Per ottimizzare il controllo degli accessi, viene adottata una matrice che rappresenta, in righe, gli attori e, in colonne, le classi. Ogni cella della matrice (attore, classe) indica le operazioni consentite per quell'attore sulle istanze della classe associata.

Sottosistema Attore	Profilo	Classe virtuale	Materiale Didattico
Docente	<ul style="list-style-type: none"> • Visualizzazione profilo • Modifica profilo anagrafico • Modifica password 	<ul style="list-style-type: none"> • Visualizzazione classe • Creazione classe • Inserimento studenti • Rimozione studenti 	<ul style="list-style-type: none"> • Visualizzare materiale • Caricare materiale • Modificare materiale • Rimuovere materiale
Studente	<ul style="list-style-type: none"> • Visualizzazione profilo • Modifica profilo anagrafico • Modifica 	<ul style="list-style-type: none"> • Visualizzazione classe 	<ul style="list-style-type: none"> • Visualizzare materiale

	password		
--	----------	--	--

Sottosistema Attore	Scenario	Dashboard	Quiz
Docente	<ul style="list-style-type: none"> • Crea scenario • Visualizza dettagli scenario • Visualizza punteggio scenario 	<ul style="list-style-type: none"> • Visualizza classifica generale • Visualizza classifica classe • Visualizza storico dello studente 	<ul style="list-style-type: none"> • Crea quiz • Visualizza storico dei quiz • visualizza risultati
Studente	<ul style="list-style-type: none"> • Visualizza dettagli scenario • Sceglie ruolo politico • Avvia scenario • Visualizza punteggio scenario 	<ul style="list-style-type: none"> • Visualizza punteggio personale • Visualizza classifica classe • Visualizza storico personale 	<ul style="list-style-type: none"> • Visualizza quiz • Esegue quiz • Visualizza risultati

3.6 Controllo flusso globale del sistema

Nel sistema BeYourChoice, ogni funzionalità è basata su interazioni da parte dell'utente, rendendolo un sistema altamente interattivo. Per questo motivo, è stato adottato un controllo del flusso globale di tipo **Event-driven**, in cui il flusso è guidato dagli eventi generati dalle azioni degli utenti.

Quando un utente interagisce con l'interfaccia, come l'accesso a quiz, la visualizzazione di materiali didattici, o la partecipazione alla chat, viene generato un evento. Questo evento viene quindi intercettato e gestito da un **dispatcher**, il quale, in base alle informazioni associate all'evento, indirizza la richiesta al sottosistema appropriato che si occupa della logica di controllo della funzionalità richiesta. Il dispatcher funge da coordinatore centrale, assicurando che ogni evento attivi il sottosistema corretto e ottimizzando così il flusso di lavoro del sistema. Questa struttura di controllo rende il sistema flessibile e scalabile, consentendo di aggiungere nuove funzionalità in futuro senza modificare le componenti principali. Inoltre, la natura event-driven del sistema lo rende particolarmente adatto per interfacce utente interattive, permettendo una gestione reattiva e fluida delle richieste e offrendo un'esperienza utente ottimizzata e intuitiva.

3.7 Condizioni Boundary

3.7.1 Avvio del sistema

Identificativo: UCB_1	Avvio del sistema	Data	15/11/23
		Versione	1.0
		Autore	Aldo Damiano Antonio Rapa
Descrizione	Lo UC definisce il processo di avvio del sistema BeYourChoice. L'avvio richiede l'inizializzazione del runtime di Python, il framework web Flask e la connessione al database MongoDB per garantire l'accesso ai dati persistenti degli utenti.		
Attore Principale	Amministratore di sistema L'amministratore di sistema vuole avviare il sistema.		
Attore Secondari	N/A		
Entry Condition	Il sistema è spento. OR Il sistema è stato riavviato. AND L'amministratore di sistema vuole avviare il sistema.		
Entry Condition On success	Il sistema è avviato e pronto all'uso.		
Exit Condition On success	Il sistema è avviato con successo e operativo.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore di sistema	Esegue il comando per avviare il server.	

2	Sistema	Il sistema inizializza il runtime di Python e il framework Flask, verifica l'integrità dei dati persistenti, si connette al database MongoDB e avvia il server.
I Scenario/Flusso di eventi Alternativo: Il server non era stato arrestato correttamente		
2.1	Sistema	Il sistema notifica l'amministratore di sistema del precedente errore nell'arresto del server e avvia il server.
II Scenario/Flusso di eventi Alternativo: I dati persistenti sono danneggiati		
2.1	Sistema	Il sistema notifica l'amministratore di sistema che sono presenti problemi relativi ai dati persistenti e non avvia il server.
2.2	Amministratore di sistema	Corregge i dati persistenti e riesegue in comando di avvio server.
2.3	Sistema	Il sistema effettua il punto 2.

3.7.2 Arresto del sistema

Identificativo: UCB_2	Arresto del sistema	Data	15/11/23
		Versione	1.0
		Autore	Aldo Damiano Antonio Rapa
Descrizione	Lo UC permette all'Amministratore di sistema di arrestare il sistema.		

Attore Principale	Amministratore di sistema L'amministratore di sistema vuole arrestare il sistema.	
Attore Secondari	N/A	
Entry Condition	Il server è correttamente avviato. AND L'amministratore di sistema ha effettuato l'accesso al sistema. AND L'amministratore di sistema vuole effettuare l'arresto del sistema.	
Entry Condition On success	Il server viene arrestato con successo.	
Exit Condition On success	Il server non viene arrestato correttamente.	
FLUSSO DI EVENTI PRINCIPALI/MAIN SCENARIO		
1	Amministratore di sistema	Esegue il comando per arrestare il server.
2	Sistema	Il sistema effettua il salvataggio dei dati persistenti, chiude le connessioni attive ed arresta il server.
I Scenario / Flusso di eventi di errore: Il salvataggio dei dati non è andato a buon fine		
2.1	Sistema	Il sistema notifica l'amministratore di sistema dell'errore e non procede con l'arresto del server.
II Scenario / Flusso di eventi di errore: Errore nella chiusura di connessioni		

2.1	Sistema	Il sistema notifica l'amministratore di sistema di errori nella chiusura di una o più connessioni attive e annulla la procedura di arresto.
2.2	Amministratore di sistema	Riesegue lo step 1.

3.7.3 Fallimento del sistema

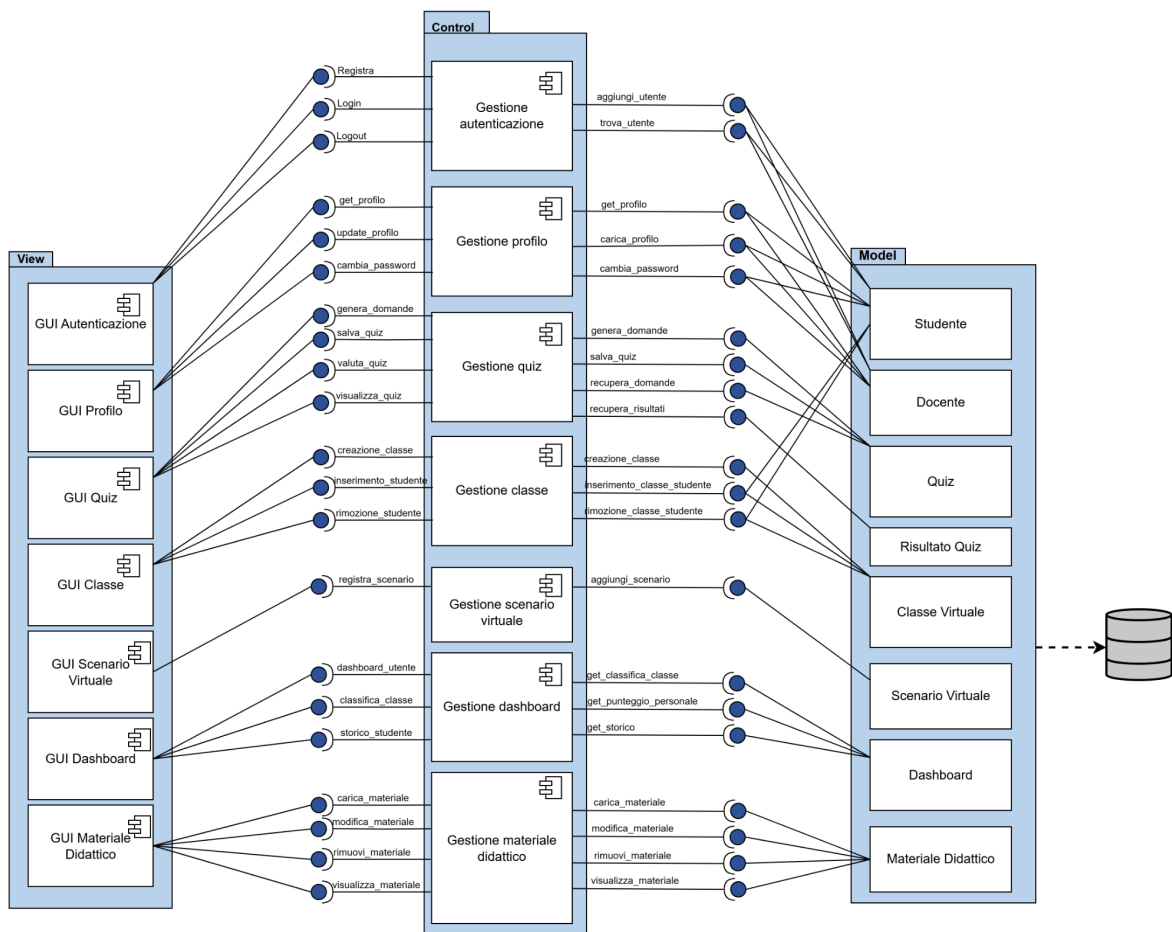
Identificativo: UCB_3	Fallimento del sistema	Data	15/11/23
		Versione	1.0
		Autore	Aldo Damiano Antonio Rapa
Descrizione	Lo UC definisce il comportamento del sistema in caso di fallimento.		
Attore Principale	Amministratore di sistema L'amministratore di sistema è interessato al comportamento del sistema in caso di errore.		
Attore Secondari	N/A		
Entry Condition	Un errore critico viene rilevato dal sistema e viene generato un allarme automatico.		
Entry Condition On success	Il Sistema viene riavviato con successo.		
Exit Condition On success	Il Sistema resta spento.		
FLUSSO DI EVENTI PRINCIPALI/MAIN SCENARIO			
1	Amministratore di sistema	L'Amministratore riceve notifica dell'errore di Sistema.	

2

Sistema

include (UCB_1) che gestisce l'avvio del Sistema.

4 Servizi di sottosistemi



4.1 GUI Autenticazione - 3 Servizi per Control

- Registra
- Login
- Logout

4.2 GUI Profilo - 3 Servizi per Control

- get_profilo
- update_profilo
- cambia_password

4.3 GUI Quiz - 4 Servizi per Control

- genera_domande
- salva_quiz
- valuta_quiz
- visualizza_quiz

4.4 GUI Classe - 3 Servizi per Control

- creazione_classe
- inserimento_studente
- rimozione_studente

4.5 GUI Scenario Virtuale - 1 Servizi per Control

- registra_scenario

4.6 GUI Dashboard - 3 Servizi per Control

- dashboard_utente
- classifica_classe
- storico_studente

4.7 GUI Materiale Didattico - 4 Servizi per Control

- carica_materiale

- modifica_materiale
- rimuovi_materiale
- visualizza_materiale

4.8 Gestione Classe - 3 Servizi per Model

- creazione_classe
- inserimento_studente
- rimozione_studente

4.9 Gestione Scenario Virtuale - 1 Servizio per Model

- aggiungi_scenario

4.10 Gestione Autenticazione - 2 Servizi per Model

- aggiungi_utente
- trova_utente

4.11 Gestione Profilo - 3 Servizi per Model

- get_profilo
- carica_profilo
- cambia_password

4.12 Gestione Quiz - 4 Servizi per Model

- genera_domande
- salva_quiz
- recupera_domande
- recupera_risultati

4.13 Gestione Classe - 3 Servizi per Model

- creazione_classe
- inserimento_classe_studente
- rimozione_classe_studente

4.14 Gestione Scenario Virtuale - 1 Servizi per Model

- aggiungi_scenario

4.15 Gestione Dashboard - 3 Servizi per Model

- get_classifica_classe
- get_punteggio_personale
- get_storico

4.16 Gestione Materiale Didattico - 4 Servizi per Model

- carica_materiale
- modifica_materiale
- rimuovi_materiale
- visualizza_materiale