System Design

Design Goals

1. **Performance**
   1. **Tempo di risposta**
      1. La validazione del Green Pass deve essere effettuata entro 2 secondi.
   2. **Throughput**
      1. Deve essere supportata la validazione contemporanea di almeno 20 Green Pass per sessione
   3. **Memoria**
      1. La quantità di memoria occupata dal sistema dipende da quella necessaria al mantenimento del database
2. **Dependability**
   1. **Robustezza**
      1. Gli input non validi inseriti dall’utente devono essere segnalati con messaggi d’errore
   2. **Affidabilità**
      1. Il sistema deve garantire che i Green Pass ricevuti per la validazione non siano presenti in copie
      2. Il report generato deve essere consistente con le informazioni contenute nei singoli esiti
   3. **Disponibilità**
      1. Il sistema deve essere disponibile durante orario universitario per il Docente e H24 per il Direttore di Dipartimento
   4. **Tolleranza agli errori**
      1. In caso di errore durante la validazione di uno o più Green Pass la sessione di validazione deve essere invalidata
   5. **Sicurezza**
      1. Tutte le password devono essere crittografate prima di essere memorizzate
      2. Il sistema deve garantire il rispetto delle leggi sulla privacy
3. **Cost** 
   1. **Costi di sviluppo**
      1. Il costo complessivo del progetto ammonta ad un massimo di 200 ore (max 50 ore per ogni membro del team)
   2. **Costi di installazione**
4. **Maintenance**
   1. **Estensibilità**
      1. Il sistema deve essere progettato in modo tale che sia possibile aggiungere moduli su richiesta del cliente
   2. **Adattabilità**
      1. Il sistema deve essere progettato su una struttura generica in modo da poter essere utilizzato in altri ambiti professionali (es. aziende, convegni, ecc…)
   3. **Portabilità**
      1. Il sistema deve essere fruibile su tutti i dispositivi mobile e desktop in maniera indipendente dal sistema operativo o dall’hardware utilizzato
   4. **Tracciabilità dei requisiti**
      1. La tracciabilità dei requisiti deve essere garantita da una matrice di tracciabilità che permette di ricondurre ogni artefatto al proprio requisito
5. **End User**
   1. **Usabilità**
      1. Il sistema deve avere un’interfaccia semplice e immediata in modo da consentire un’interazione rapida e efficiente, così da ridurre i tempi di controllo

**Trade-off**

1. Tempo di rilascio vs funzionalità

Per rispettare le scadenze del Progetto potrebbe essere necessaria l’implementazione parziale di alcune delle funzionalità richieste.

1. Usabilità vs memoria

Poichè uno degli obiettivi del Sistema è l’alta usabilità per gli utenti finali, è necessario memorizzare una grande porzione di dati e ciò va a discapito di una più elevata dimensione del database.

1. Velocità vs memoria

Per favorire un rapido accesso ai dati risulta essere più vantaggioso utilizare una quantità di memoria più elevate.

**Architettura del Sistema corrente**

Non c’è un Sistema corrente.

**Architettura del Sistema proposto**

Il Sistema proposto è una Web Application che vuole facilitare, velocizzare e rendere più sicuro il controllo dei Green Pass degli studenti, effettuato dai docenti in aula. Poichè l’usabilità è uno degli obiettivi principali del Sistema, è d’obbligo fornire una facile interazione con il Sistema. Pertanto, la scelta dell’architettura è ricaduta sul modello MVC, appunto solitamente selezionato per la realizzazione di sistemi interattivi.

Questo pattern architetturale si compone di tre layer:

* **Model**: si occupa di mantenere le informazioni del dominio applicativo;
* **View**: permette la visualizzazione delle informazioni fornite dal Model;
* **Controller**: è responsabile delle interazioni con l’utente, recuperando i dati dal model mandandoli alle view per la visualizzazione.

**Decomposizione del Sistema in sottosistemi**

Il Sistema si compone in totale di 14 componenti, suddivisi tra i layer View, Model e Controller.

Il **Controller** contiene le componenti:

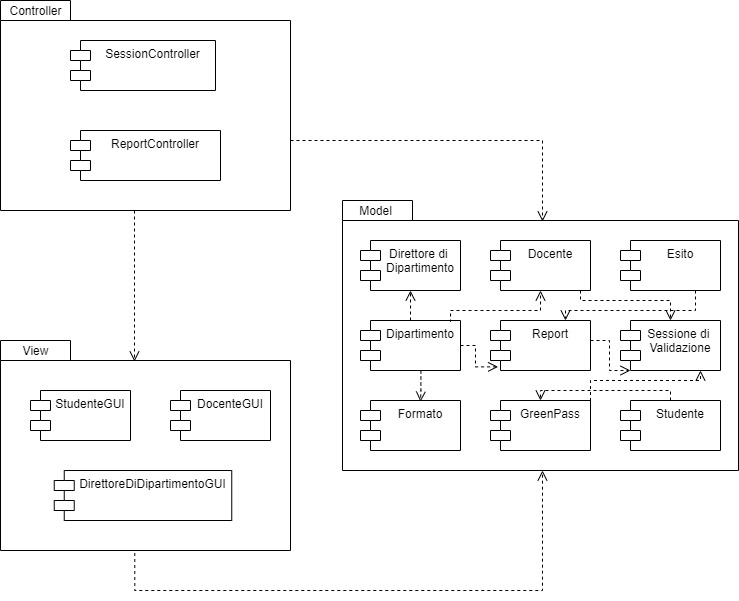
* **SessionController**: Si occupa di tutte le operazioni che concernono lo svolgimento di una sessione di validazione: dall’avvio della sessione, alla ricezione e validazione dei Green Pass, fino alla terminazione.
* **ReportController**: Si occupa di gestire tutte le operazioni che possono essere effettuate dal Direttore di Dipartimento sui report, quindi ricerca, download, cancellazione e selezione del formato.

Il **Model** contiene le componenti:

* **Direttore** **di** **Dipartimento**: Modella le caratteristiche di un Direttore di Dipartimento e consente di interfacciarsi con l’entità nel database per compiere le operazioni CRUD.
* **Docente**: Modella le caratteristiche di un Docente e consente di interfacciarsi con l’entità nel database per compiere le operazioni CRUD.
* **Esito**: Modella le informazioni relative all’esito della validazione di un GreenPass e si occupa dlel’interfacciamento con il database per compiere le operazioni CRUD.
* **Dipartimento**: Modella un Dipartimento e le sue caratteristiche principali.
* **Report**: Modella un report di una validazione e consente di interfacciarsi con la base di dati per compiere le operazioni CRUD.
* **Sessione** di Validazione: Modella una Sessione di Validazione conservandone le proprietà e consente di interfacciarsi con il database per compiere le operazioni CRUD.
* **Formato**: Si occupa di mantenere le proprietà di un Formato e si interfaccia con il database per compiere le operazioni CRUD.
* **GreenPass**: Modella un GreenPass.
* **Studente**: Modella uno Studente mantenendo le principali proprietà.

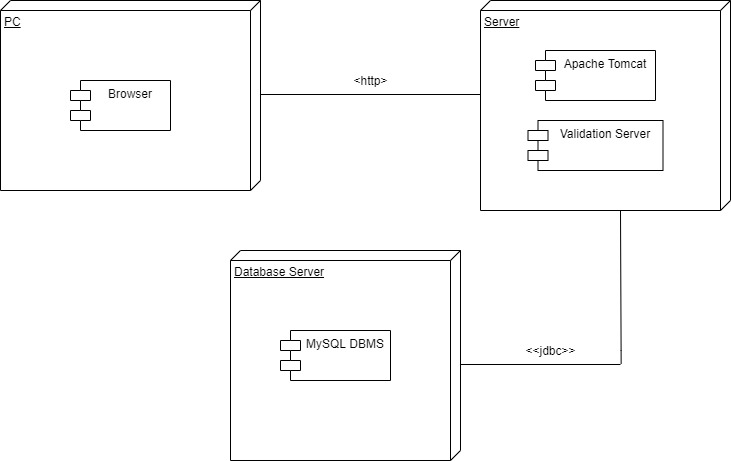
Il **View** contiene le componenti:

* **StudenteGUI:** interfaccia grafica per lo Studente.
* **DocenteGUI:** interfaccia grafica per il Docente.
* **DirettoreDiDipartimentoGUI:** interfaccia grafica per il Direttore di Dipartimento.



**Mapping Hardware/Software**

Il sistema EasyPass è realizzato come una Web Application. Per poter interagire con il Sistema è necessario accedervi tramite un Browser da un dispositivo qualsiasi. Il Web Browser comunicherà con in Web Server Apache Tomcat attraverso il protocollo HTTP, che si occuperà pertanto di elaborare e rispondere a richieste dal client. La persistenza dei dati è invece mantenuta tramite l’utilizzo di un Database MYSQL, che sarà contattato dal Server tramite JDBC. Oltre al Web Server sarà necessario l’utilizzo di un Validation Server, il quale assolverà la funzione di validazione dei Green Pass e che sarà in comunicazione con il Web Server sempre tramite richieste e risposte http.



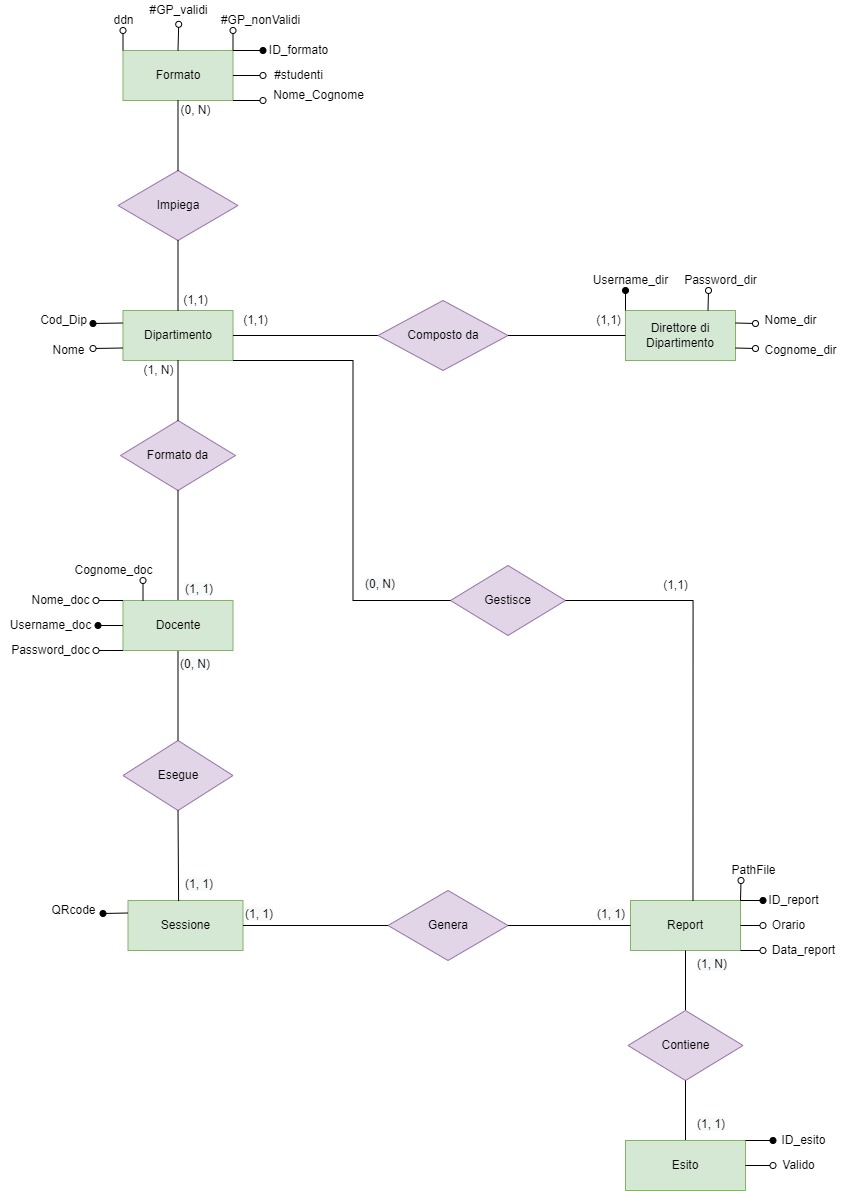
**Gestione dei dati persistenti**

Per la memorizzazione dei dati persistenti si è deciso di utilizzare un RDBMS (Relational Database

Management System) poiché permette di accedere in modo semplice ed efficiente ai dati, conservandone la consistenza, la privatezza e l'affidabilità. Possiamo effettuare ricerche complesse (ad esempio, la ricerca di report sulla base di un filtro scelto) che, se compiute su tradizionali archivi analogici, comporterebbero ampio dispendio di tempo e risorse.

Inoltre, fornisce un accesso concorrente ai dati mantenendone la coerenza anche in condizione di multiutenza e, soprattutto, possiede un meccanismo di permessi per cui utenti con operazioni diverse possono accedere a sezioni diverse della base di dati in maniera protetta.

Di seguito si allega lo schema concettuale e del database che si intende utilizzare per la realizzazione del sistema.



|  |
| --- |
| **Regole di vincolo** |
| **(RV1)** Un formato di report composto dalla data di nascita dello studente DEVE essere composto anche dal rispettivo nome e cognome. |
| **(RV2)** La password di un direttore di dipartimento deve essere almeno di 8 caratteri. |
| **(RV3)** La password di un di docente deve essere almeno di 8 caratteri. |

|  |
| --- |
| **Vincoli di integrità referenziale** |
| **(VIF1)** La chiave esterna “*ID\_formato*” della tabella “*Dipartimento*” ha un vincolo di integrità referenziale con la chiave primaria “*ID\_formato*” della tabella “*Formato*”. |
| **(VIF2)** La chiave esterna “Codice\_dip” della tabella “*Docente*” ha un vincolo di integrità referenziale con la chiave primaria “Codice\_dip” della tabella “*Dipartimento*”. |
| **(VIF3)** La chiave esterna “*Codice\_dip*” della tabella “*Direttore*” ha un vincolo di integrità referenziale con la chiave primaria “*Codice\_dip*” della tabella “*Dipartimento*”. |
| **(VIF4)** La chiave esterna “*Username\_doc*” della tabella “*Sessione*” ha un vincolo di integrità referenziale con la chiave primaria “*Username\_doc*” della tabella “*Docente*”. |
| **(VIF5)** La chiave esterna “*Codice\_dip*” della tabella “*Report*” ha un vincolo di integrità referenziale con la chiave primaria “*Codice\_dip*” della tabella “*Dipartimento*”. |
| **(VIF5)** La chiave esterna “*QRcode\_session*” della tabella “*Report*” ha un vincolo di integrità referenziale con la chiave primaria “*QRcode*” della tabella “Sessione”. |
| **(VIF7)** La chiave esterna “*ID\_report*” della tabella “*Esito*” ha un vincolo di integrità referenziale con la chiave primaria “*ID\_report*” della tabella “*Report*”. |

**Access Security Control**

La sicurezza per l’accesso al sistema è garantita tramite autenticazione con credenziali private. A seconda di se si tratta di un direttore di dipartimento o di un docente si sarà reindirizzati a pagine diverse, assicurando la legittimità delle operazioni compiute.

Ciascun direttore ha delle credenziali che gli sono state assegnate e che saranno passate al direttore successivo nel momento in cui finisce il suo mandato; al contrario. i docenti dovranno registrarsi in autonomia.

In particolare, un docente potrà eseguire rilevamenti e creare report, eventualmente scaricandoli sul proprio dispositivo. Un direttore avrà accesso a tutti i report registrati nel suo dipartimento con la possibilità di svolgere operazioni di ricerca, formattazione, eliminazione e salvataggio dei report.

L’unica possibilità di interazione col sistema data allo studente, invece, consiste nell’invio del GP. Per motivi di privacy, dopo che l’esito della valutazione del GP è stato registrato, il GP stesso sarà cancellato, senza essere salvato nel database.

**È evidenziato perchè sarà inserito nella matrice ACL**

**Controllo del flusso globale del software**

Il sistema può essere definito come “event-driven” perchè abbiamo un ciclo principale che aspetta il verificarsi di un evento esterno. Ad esempio,

**Condizione limite**

**Avvio del sistema**. Al momento dell’avvio, il sistema necessita di un web server che fornisca il servizio di

accesso ad un database MySQL per la gestione dei dati persistenti e l’interpretazione ed esecuzione del codice lato server. Quando un utente (docente, direttore di dipartimento o studente) accede al sistema, gli verrà presentata una pagina nel browser web con le operazioni che gli è consentito svolgere a seconda del suo ruolo all’interno del sistema.

**Terminazione**. Alla chiusura dell’applicazione, il sistema termina con un logout automatico, laddove necessario. Il sistema non si occupa di salvare eventuali dati lasciati in sospeso durante l’utilizzo, sia che l’utente si trovasse nel lato amministrativo, sia che si trovasse nel lato utente. Pertanto, al riavvio del sistema, quest’ultimo non ripresenterà i dati immessi in precedenza se questi non sono stati opportunamente salvati. Il server dovrà essere terminato manualmente dall’amministratore del sistema, dopodiché nessun client potrà connettersi alla piattaforma.

**Fallimento**. Si possono individuare diverse situazioni di fallimento:

* Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso non è prevista nessuna procedura di salvataggio né di ripristino dei dati. //DA RIVEDERE
* Nel caso in cui si verifichi un'interruzione inaspettata dell’alimentazione, non sono previste procedure in grado di riportare il sistema nello stato in cui era prima dello spegnimento.
* Nel caso di un fallimento del software stesso con conseguente chiusura inaspettata dovuta ad errori commessi durante la fase di validazione, non si prevedono politiche di recupero specifiche, se non il riavvio dell’intero sistema nel caso di errori fatali.
* Nel caso in cui si verifichi un errore nell’hardware (ad es. un dispositivo di archiviazione di massa), non si prevedono contromisure software.