



Laurea Triennale in Informatica - Università di Salerno
Corso di *Ingegneria del Software* - Prof C. Gravino



System Design Document Easy Pass

Riferimento	
Versione	2.1
Data	15/12/2021
Destinatario	Prof. C. Gravino
Presentato da	Montefusco Alberto Mulino Martina Rinaldi Viviana Spina Gennaro
Approvato da	



Sommario

Sommario	2
Revision History	3
Responsabilità	4
1. Introduzione	5
1.1 Scopo del Sistema	5
1.2 Obiettivi di Design	5
1.2.1 Design Trade-off	7
1.3 Definizioni, acronimi e abbreviazioni	7
1.4 Riferimenti	8
1.5 Panoramica	9
2. Architettura del Sistema Corrente	9
3. Architettura del Sistema Proposto	9
3.1 Panoramica	10
3.2 Decomposizione in Sottosistemi	10
3.3 Mapping Hardware/Software	12
3.4 Gestione dei dati persistenti	13
3.4.1 Dizionario dei dati	14
3.5 Controllo degli accessi e sicurezza	18
3.6 Controllo del flusso globale del Software	18
3.7 Condizione limite	19
4. Servizi dei Sottosistemi	20
5. Glossario	22



Revision History

Data	Versione	Descrizione	Autori
19/11/2021	0.1	Prima stesura: design goals, trade-off, architettura del sistema proposto.	Alberto Montefusco Gennaro Spina Viviana Rinaldi Martina Mulino
22/11/2021	0.2	Aggiunta "Controllo degli accessi e sicurezza". Aggiunta "Gestione dei dati persistenti" con relativo modello concettuale, regole di vincolo e creazione del database in MySQL.	Viviana Rinaldi Alberto Montefusco
23/11/2021	0.3	Aggiunta "Condizione limite" e "Gestione del flusso globale del Sistema". Aggiunta del modello MVC, deployment diagram, component diagram, divisione del sistema in sottosistemi e mapping Hardware/Software.	Alberto Montefusco Viviana Rinaldi Martina Mulino
24/11/2021	0.4	Aggiunta dei servizi offerti dal sottosistema.	Viviana Rinaldi
29/11/2021	1.0	Revisione SDD.	Alberto Montefusco Gennaro Spina Viviana Rinaldi Martina Mulino
30/11/2021	1.1	Eliminazione delle componenti "Green Pass" e "Studente"	Alberto Montefusco Gennaro Spina Viviana Rinaldi Martina Mulino



9/12/2021	2.0	Revisione finale SDD	Alberto Montefusco Gennaro Spina Viviana Rinaldi Martina Mulino
15/12/2021	2.1	Modifica Servizi dei Sottosistemi: aggiunta "AutenticazioneGUI" e modifica "DocenteGUI" e "DirettoreDiDipartimentoGUI"	Alberto Montefusco Gennaro Spina Viviana Rinaldi Martina Mulino

Responsabilità

Artefatto	Autori
Design Goals, trade-off	Alberto Montefusco, Gennaro Spina, Viviana Rinaldi, Martina Mulino
Architettura del Sistema corrente e del Sistema proposto	Martina Mulino
Decomposizione del Sistema in sottosistemi con relativi diagrammi UML	Martina Mulino
Mapping Hardware/Software	Martina Mulino
Gestione dei dati persistenti	Alberto Montefusco
Controllo degli accessi e sicurezza	Viviana Rinaldi
Controllo del flusso globale del Software e condizione limite	Alberto Montefusco, Viviana Rinaldi
Servizi offerti dai sottosistemi	Viviana Rinaldi



1. Introduzione

1.1 Scopo del Sistema

Come meglio illustrato nel paragrafo 1.1 del documento “RAD”, la realizzazione di Easy Pass ha come obiettivo quello di essere di supporto ai Docenti nel compiere le loro operazioni di controllo sugli Studenti. In particolare, con questo progetto, la validazione dei Green Pass degli Studenti risulterà più rapida, efficiente e sicura.

1.2 Obiettivi di Design

Di seguito, vengono descritti gli obiettivi di design che devono essere rispettati nella realizzazione del Sistema. Tali obiettivi sono organizzati in cinque categorie:

1. Criteri di performance

- **Tempo di risposta**

- La validazione del Green Pass deve essere effettuata entro 2 secondi.

- **Throughput**

- Deve essere supportata la validazione contemporanea di almeno 20 Green Pass per sessione.

- **Memoria**

- La quantità di memoria occupata dal Sistema dipende da quella necessaria al mantenimento del database.

2. Criteri di dependability

- **Robustezza**

- Gli input non validi inseriti dall'utente devono essere segnalati con messaggi d'errore.

- **Affidabilità**

- Il Sistema deve garantire che i Green Pass ricevuti per la validazione non siano presenti in copie.
- Il report generato deve essere consistente con le informazioni contenute nei singoli esiti.



- **Disponibilità**
 - Il Sistema deve essere disponibile durante orario universitario per il Docente e H24 per il Direttore di Dipartimento.
- **Tolleranza agli errori**
 - In caso lo studente invii un'immagine che non sia il GP, gli viene offerta la possibilità di ripetere l'operazione e viene impedita la trasmissione del file errato al server di validazione.
- **Sicurezza**
 - Tutte le password devono essere crittografate prima di essere memorizzate.
 - Il Sistema deve garantire il rispetto delle leggi sulla privacy, memorizzando soltanto le informazioni consentite dalle normative stesse.
- 3. **Criteri di costo**
 - **Costi di sviluppo**
 - Il costo complessivo del progetto ammonta ad un massimo di 200 ore (max 50 ore per ogni membro del team).
- 4. **Criteri di manutenzione**
 - **Estensibilità**
 - Il Sistema deve essere progettato in modo tale che sia possibile aggiungere moduli su richiesta del cliente.
 - **Adattabilità**
 - Il Sistema deve essere progettato su una struttura generica in modo da poter essere utilizzato in altri ambiti professionali (es. aziende, convegni, ecc.).
 - **Portabilità**
 - Il Sistema deve essere fruibile su tutti i dispositivi mobile e desktop in maniera indipendente dal sistema operativo o dall'hardware utilizzato.
 - **Tracciabilità dei requisiti**
 - La tracciabilità dei requisiti deve essere garantita da una matrice di tracciabilità che permette di ricondurre ogni artefatto al proprio requisito.
- 5. **Criteri dell'utente finale**
 - **Usabilità**
 - Il Sistema deve avere un'interfaccia semplice e immediata in modo da consentire un'interazione rapida ed efficiente, così da ridurre i tempi di controllo.



1.3 Design Trade-off

Tempo di rilascio Vs funzionalità

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

Usabilità Vs memoria

Poiché uno degli obiettivi del Sistema è l'alta usabilità per gli utenti finali, è necessario memorizzare una grande porzione di dati e ciò porta ad un elevato uso della memoria del database.

Velocità Vs memoria

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

1.4 Definizioni, acronimi e abbreviazioni

In questa sezione descriveremo i termini che sono stati utilizzati all'interno del Documento stesso divisi in tre sezioni principali: definizioni, acronimi ed abbreviazioni.

1. Definizioni:

- **COVID-19:** nome di una malattia infettiva respiratoria causata dal virus denominato SARS-CoV-2 appartenente alla famiglia dei Coronavirus;
- **Green Pass:** nome con il quale indichiamo un certificato interoperabile all'interno dell'Unione europea, contenente le informazioni che attestano che il titolare è stato vaccinato contro la COVID-19, o ha da poco effettuato un test diagnostico per SARS-CoV-2 con risultato negativo, oppure è guarito dalla COVID-19;
- **Esito:** nome con cui indichiamo il risultato della validazione (positivo/negativo) del Green Pass;
- **Report:** nome con il quale ci si riferisce all'insieme degli esiti raccolti durante una lezione tenuta da un Docente;



- **Sessione di validazione:** nome con il quale indichiamo la durata temporale in cui viene effettuata una validazione;
- **Validation Server:** server creato appositamente per la validazione dei Green Pass tramite API fornite dal Ministero della Salute.

2. Acronimi:

- **RV[n]:** Regole di Vincolo[numero];
- **VIR[n]:** Vincolo di Integrità Referenziale[numero];
- **RDBMS:** Relational Database Management System (gestore di database basati sul modello relazionale);
- **DB:** DataBase, ovvero “Base di Dati” utilizzata per memorizzare i dati persistenti;
- **CRUD:** Create, Read, Update e Delete. Queste quattro parole fanno riferimento alle quattro principali operazioni che si svolgono su un database relazionale, ovvero la creazione di tabelle, dati e relazioni (create), la loro lettura (read), la loro modifica tramite aggiornamento (update) o eliminazione (delete);
- **SDD:** System Design Document.

3. Abbreviazioni:

- **GP:** Green Pass;
- **Direttore:** si intende il Direttore di Dipartimento.

1.5 Riferimenti

Per stilare la presente documentazione, si è preso come riferimento le slide fornite dal Docente del corso di Ingegneria del Software, Carmine Gravino, inserite nella sezione “M3” della piattaforma di e-learning della facoltà di Informatica. Inoltre, è stato consultato il libro di testo “Object-Oriented Software Engineering Using UML, Patterns and Java: Third Edition, di Bernd Bruegge ed Allen H. Dutoit”.

1.6 Panoramica

Dopo questa prima sezione di introduzione del presente Documento, il punto 2 descriverà brevemente l'architettura del Sistema corrente, mentre al punto 3 verrà fornita una dettagliata descrizione



dell'architettura del Sistema proposto. In particolare, questa sezione descriverà la decomposizione in sottosistemi, la corrispondenza tra hardware e software, la gestione dei dati persistenti, il controllo degli accessi, la sicurezza, il flusso di controllo e la gestione delle condizioni limite del Sistema proposto. Infine, il punto 4 descriverà i servizi offerti dai sottosistemi individuati.

2. Architettura del Sistema corrente

Il Sistema che si vuole realizzare è il soggetto di un progetto Greenfield, sviluppato a partire da una situazione reale non informatizzata. Pertanto, non è possibile descrivere l'architettura del meccanismo correntemente utilizzato.

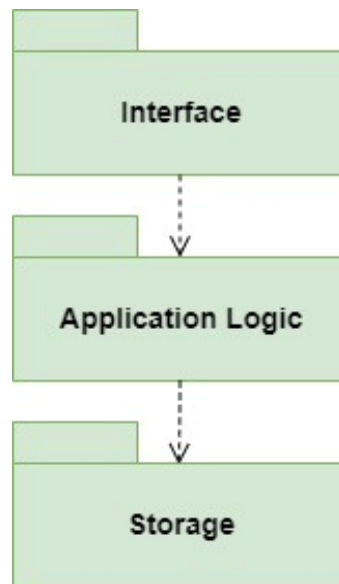
3. Architettura del Sistema proposto

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. L'architettura software scelta per la realizzazione del Sistema è la Three Tier, la quale, fornendo la possibilità di eseguire ciascun tier sulla propria infrastruttura, offre numerosi vantaggi, tra cui uno sviluppo più veloce e una maggiore scalabilità, affidabilità e sicurezza.

Questo pattern architetturale si compone di tre livelli:

1. **Interface:** è il tier di presentazione, dunque delle interfacce utente. Si occupa di visualizzare le informazioni all'utente e di raccogliere informazioni da quest'ultimo;
2. **Application Logic:** è il tier della logica di business dell'applicazione. Si occupa di elaborare le informazioni raccolte nel tier Interface e di aggiungere, rimuovere o modificare i dati nel tier Storage;
3. **Storage:** è il tier dell'accesso ai dati. Si occupa della persistenza delle informazioni e della loro gestione e archiviazione.

Nel Sistema realizzato con la suddetta architettura tutte le comunicazioni passano attraverso l'Application Logic tier; l'Interface tier e lo Storage tier non comunicano mai direttamente tra loro.



3.1 Panoramica

Di seguito, vengono illustrati i risultati della fase di progettazione del Sistema proposto. Viene descritta l'architettura del sistema, stabilita in base alla struttura del sito web e alle funzionalità che deve offrire. Particolare attenzione è stata riposta nella gestione dei dati persistenti, i quali risultano fondamentali allo scopo del sistema proposto.

3.2 Decomposizione in Sottosistemi

Il Sistema si compone in totale di 14 componenti, suddivisi tra i layer Interface, Application Logic e Storage.

L'**Application Logic** layer contiene le componenti:

- **AccessController**: si occupa di tutte le operazioni che riguardano il controllo degli accessi degli utenti che utilizzano il Sistema, come ad esempio il login del Direttore e del Docente;
- **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.

Lo **Storage** layer contiene le componenti:

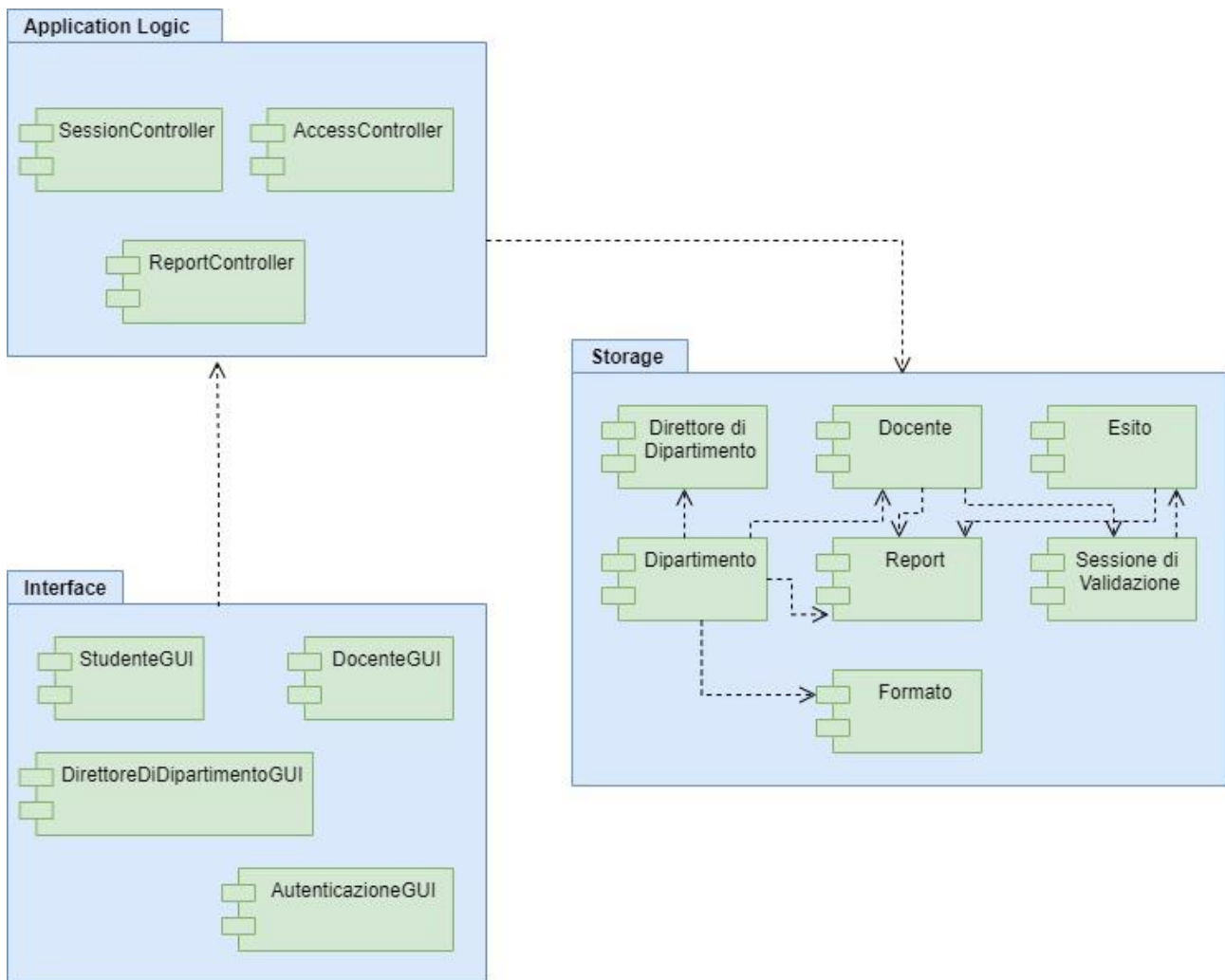
- **Direttore di Dipartimento**: modella le caratteristiche di un Direttore di Dipartimento e consente di interfacciarsi con l'entità nel database per compiere le operazioni CRUD;
- **Docente**: modella le caratteristiche di un Docente e consente di interfacciarsi con l'entità nel database per compiere le operazioni CRUD;
- **Esito**: modella le informazioni relative all'esito della validazione di un Green Pass e si occupa dell'interfacciamento con il database per compiere le operazioni CRUD. L'esito, in particolare, contiene i dati dello Studente che ha sottomesso il Green Pass e la stringa che identifica quest'ultimo, oltre che la dicitura che indica il risultato della validazione.

NB: Nel rispetto del requisito non funzionale RNF[3] (vedi documento "RAD" a pag. 15), nel database non verrà memorizzato l'attributo che contiene la stringa rappresentante il Green Pass.

- **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;

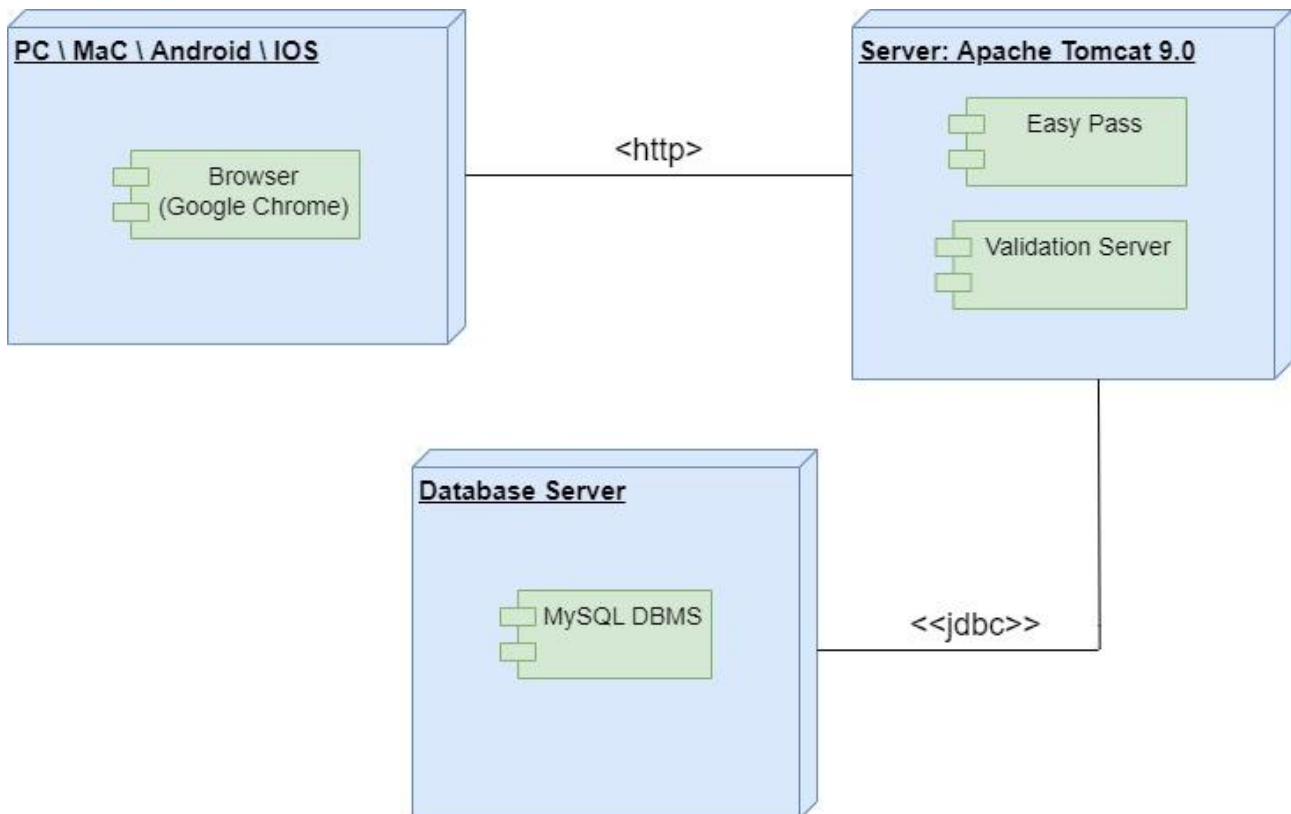
L'**Interface** layer contiene le componenti:

- **StudenteGUI**: interfaccia grafica per lo Studente;
- **DocenteGUI**: interfaccia grafica per il Docente;
- **AutenticazioneGUI**: interfaccia grafica per gestire il login e la registrazione;
- **DirettoreDiDipartimentoGUI**: interfaccia grafica per il Direttore di Dipartimento.



3.3 Mapping Hardware/Software

Il sistema Easy Pass è realizzato come una Web Application. Per poter interagire con il Sistema è necessario accedervi tramite Google Chrome in quanto il Sistema è “Chrome first” da dispositivi quali PC, MAC, Android e IOS. Il Web Browser comunicherà con 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 sfruttando le API messe a disposizione dal Ministero della Salute e che sarà in comunicazione con Tomcat sempre tramite richieste e risposte HTTP.



3.4 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 privacy 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 del database che si intende utilizzare per la realizzazione del Sistema, le regole di vincolo, i vincoli di integrità referenziali e il dizionario dei dati.



3.4.1 Dizionario dei dati

Entità	Descrizione	Attributi	Identificatore
Formato	Rappresenta come è organizzato il contenuto del report.	<ul style="list-style-type: none">- Ddn- GP_validi- GP_nonValidi- N_studenti- Nome_Cognome	ID_formato
Dipartimento	Rappresenta una struttura organizzativa all'interno dell'Università.	<ul style="list-style-type: none">- Nome	Codice_Dip
Direttore di Dipartimento	È il capo di un Dipartimento.	<ul style="list-style-type: none">- Password_Dir- Nome_Dir- Cognome_Dir	Username_Dir
Docente	Responsabile dell'avvio e della terminazione delle sessioni, volte al controllo e alla validazione dei GP.	<ul style="list-style-type: none">- Password_Doc- Nome_Doc- Cognome_Doc	Username_Doc
Sessione	Rappresenta l'attività di validazione dei Green Pass svolta dal Docente.	<ul style="list-style-type: none">- isInCorso	QRcode
Report	Insieme di esiti risultanti da una Sessione.	<ul style="list-style-type: none">- Orario- Data_report- PathFile	ID_report
Esito	Rappresenta il risultato della validazione del Green Pass di uno Studente.	<ul style="list-style-type: none">- Valido- Nome_Studente- Cognome_Studente- Ddn_Studente	ID_Esito



Relazione	Descrizione	Entità coinvolte
Impiega	Associa ciascun Formato ad un solo Dipartimento.	Formato (1, 1) Dipartimento (1, 1)
Dirige	Associa un Dipartimento ad un Direttore di Dipartimento.	Dipartimento (1, 1) Direttore di Dipartimento (1, 1)
Formato da	Associa un Dipartimento ad un Docente.	Dipartimento (1, N) Docente (1, 1)
Gestisce	Associa un Dipartimento ad un Report.	Dipartimento (0, N) Report (1, 1)
Esegue	Associa un Docente ad una Sessione.	Docente (0, N) Sessione (1, 1)
Genera	Associa una Sessione ad un Esito.	Sessione (1, N) Esito (1, 1)
Contiene	Associa un Esito ad un Report.	Report (1, N) Esito (0, 1)
Produce	Associa un Report a un Docente.	Report (1, 1) Docente (0, N)



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 e contenente almeno un carattere speciale, uno maiuscolo, uno minuscolo ed almeno un numero.

(RV3) La password di un di Docente deve essere almeno di 8 caratteri e contenente almeno un carattere speciale, uno maiuscolo, uno minuscolo ed almeno un numero.

Modello Logico

Formato (ID_formato, N_studenti, GP_validi, GP_nonValidi, Nome_Cognome, Ddn)

Dipartimento (Codice_Dip, Nome, ID_formato)

Docente (Username_Doc, Nome_Doc, Cognome_Doc, Password_Doc, Codice_Dip)

Direttore (Username_Dir, Nome_Dir, Cognome_Dir, Password_Dir, Codice_Dip)

Sessione (Qrcode, isInCorso, Username_Doc)

Report (ID_report, Orario, Data_report, PathFile, Codice_Dip, Username_Doc)

Esito (ID_Esito, Valido, Nome_Studente, Cognome_Studente, Ddn_Studente, ID_Report*, QrcodeSession)

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*".

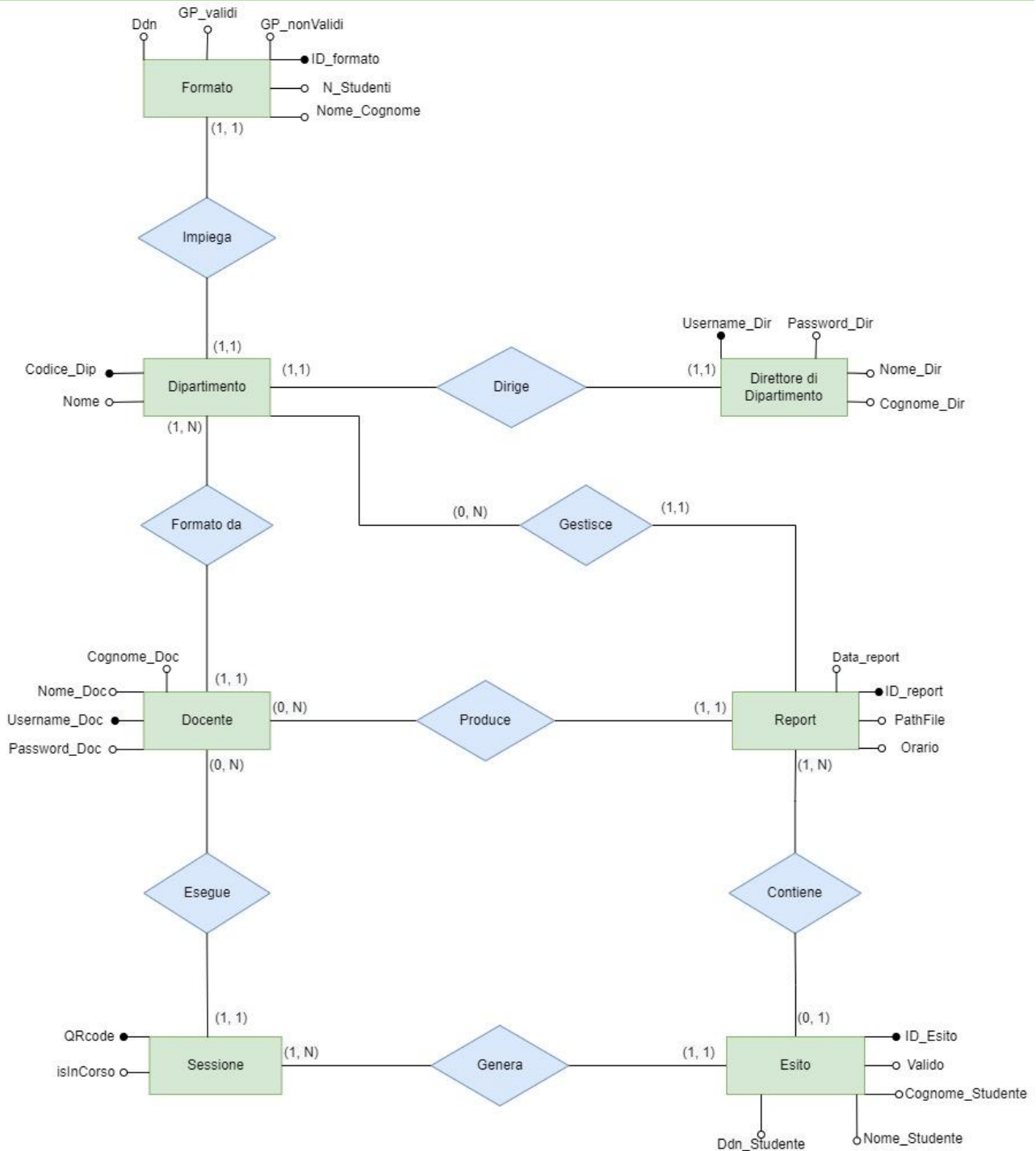
(VIF4) La chiave esterna "Username_doc" della tabella "*Sessione*" ha un vincolo di integrità referenziale con la chiave primaria "Username_doc" della tabella "*Docente*".

(VIF5) La chiave esterna "Codice_dip" della tabella "*Report*" ha un vincolo di integrità referenziale con la chiave primaria "Codice_dip" della tabella "*Dipartimento*".

(VIF6) La chiave esterna "Username_Doc" della tabella "*Report*" ha un vincolo di integrità referenziale con la chiave primaria "Username_Doc" della tabella "*Docente*".

(VIF7) La chiave esterna "*ID_report*" della tabella "*Esito*" ha un vincolo di integrità referenziale con la chiave primaria "*ID_report*" della tabella "*Report*".

(VIF8) La chiave esterna "*QRcodeSessione*" della tabella "*Esito*" ha un vincolo di integrità referenziale con la chiave primaria "*QRcodeSessione*" della tabella "*Sessione*".



3.5 Controllo degli accessi e sicurezza

La sicurezza per l'accesso al Sistema è garantita tramite autenticazione con credenziali personali. Ciascun Direttore ha delle credenziali che gli sono state assegnate al momento della nomina e che saranno passate al Direttore successivo nel momento in cui finisce il suo mandato; al contrario, i Docenti dovranno registrarsi in autonomia.

Per motivi di privacy il GP non viene mai salvato nel database.

Nella matrice di accesso che segue riportiamo:

- nella prima colonna, gli attori del Sistema;
- nella prima riga, un'istanza delle classi del nostro Sistema;
- nelle celle risultanti dall'incrocio tra riga e colonna, abbiamo le operazioni che l'attore può compiere sull'oggetto in questione.

	Sessione	Report	Dipartimento	Formato	Esito
Direttore		1. Visualizza report 2. Download report 3. Ricerca report 4. Elimina report	Visualizza dipartimento	Scelta formato	
Docente	1. Avvia Sessione 2. Termina Sessione	1. Visualizza report 2. Download report 3. Crea Report	Visualizza dipartimento		Vedi esito
Studente	Invia Green Pass				

3.6 Controllo del flusso globale del Software

Il Sistema può essere definito come “event-driven” perché abbiamo un ciclo principale che aspetta il verificarsi di un evento esterno. Ad esempio, il Sistema rimane in attesa della ricezione di un certo numero di Green Pass che dovranno essere inviati dagli Studenti.



3.7 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.

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 alcuna procedura di salvataggio né di ripristino dei dati;
- 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;
- Se un utente invia al Sistema informazioni errate oppure l'utente non sottomette delle informazioni per la corretta esecuzione di un'operazione, il Sistema risponderà con un messaggio di errore;
- Nel caso in cui si verifichi un errore nell'hardware (ad es. un dispositivo di archiviazione di massa), non si prevedono soluzioni software.



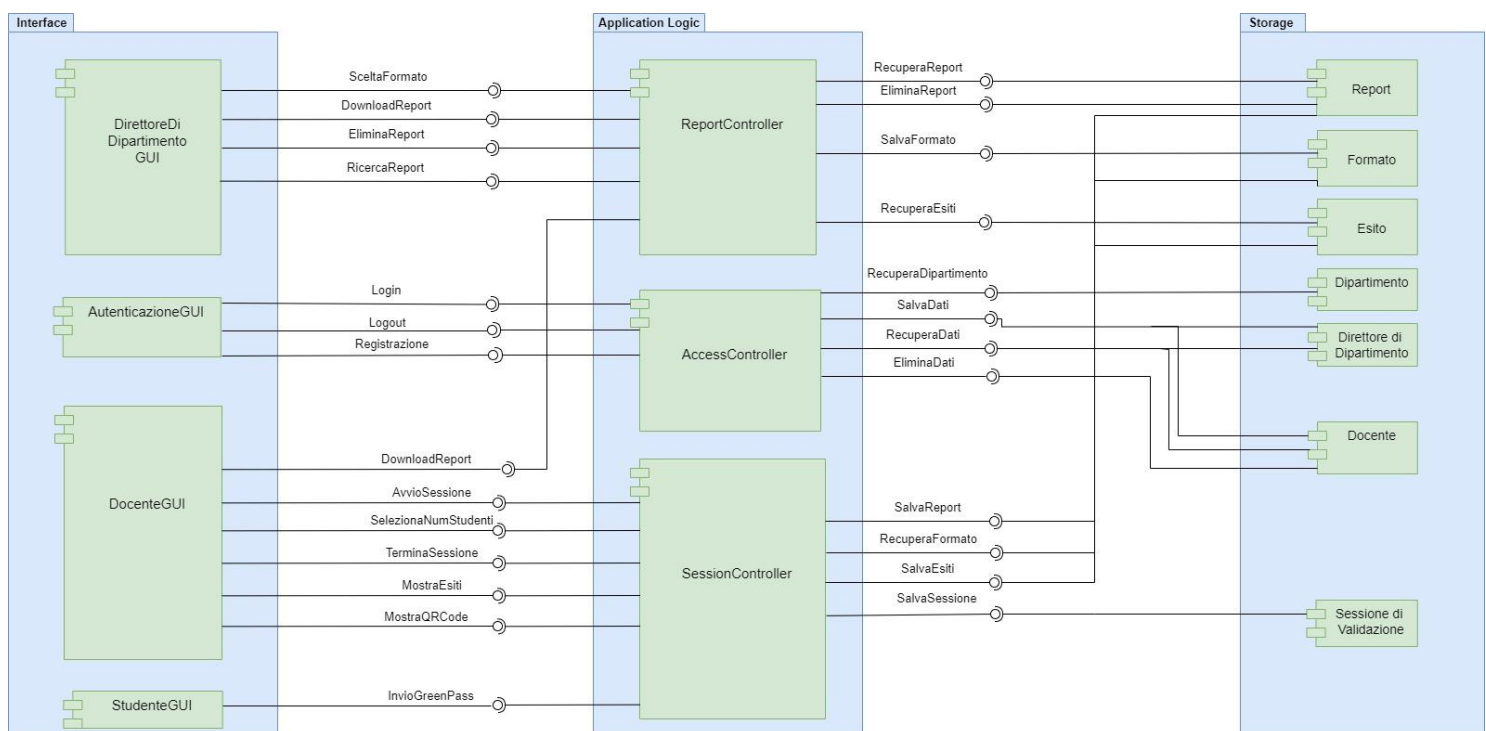
4. Servizi dei Sottosistemi

Interface:

- **StudenteGUI:**
 - **Invio Green Pass:** permette allo Studente di inviare il proprio GP.
- **DocenteGUI:**
 - **Avvio Sessione:** permette l'avvio della sessione di validazione;
 - **Mostra QR code:** permette al Docente di ottenere un QR code da mostrare agli Studenti per procedere con il controllo della validità dei loro GP;
 - **Seleziona Num Studenti:** permette al Docente di inserire il numero di Studenti di cui si vuole verificare il GP;
 - **Download Report:** permette il download dei report sulla macchina usata del Docente;
 - **Termina Sessione:** permette l'interruzione della sessione prima del raggiungimento del numero di Studenti inizialmente richiesto;
 - **Mostra Esiti:** permette al Docente di verificare l'esito dei GP ricevuti.
- **DirettoreDiDipartimentoGUI**
 - **Ricerca Report:** permette al Direttore di effettuare una ricerca tra i report del suo Dipartimento mediante opportuni filtri;
 - **Elimina Report:** permette l'eliminazione di report salvati nel DB;
 - **Download Report:** permette il download di report sulla macchina usata dal Direttore;
 - **Scelta Formato:** permette la scelta del formato desiderato del report.
- **AutenticazioneGUI**
 - **Login:** permette al Direttore e al Docente di autenticarsi tramite credenziali;
 - **Logout:** permette al Direttore e al Docente di disconnettersi;
 - **Registrazione:** permette al Docente di registrarsi al Sistema.

Application Logic:

- **SessionController:**
 - **Salva Sessione:** salva i dati della sessione nel DB;
 - **Recupera Formato:** restituisce i dati del formato scelto dal Direttore;
 - **Salva Report:** salva nel DB i dati del report appena ottenuto;
 - **Salva Esiti:** salva gli esiti delle validazioni nel DB.
- **ReportController:**
 - **Recupera Report:** restituisce al Direttore i dati dei report effettuati nel suo Dipartimento;
 - **Elimina Report:** su richiesta del Direttore, si effettua la cancellazione del report dal DB;
 - **Salva Formato:** salva la modalità di formato favorita dal Direttore;
 - **Recupera Esiti:** restituisce gli esiti registrati nel DB.
- **AccessController:**
 - **Salva Dati:** salva nel DB i dati della registrazione del Docente e del Direttore;
 - **Recupera Dati:** restituisce i dati del Docente o del Direttore;
 - **Recupera Dipartimento:** restituisce il Dipartimento di appartenenza;
 - **Elimina Dati:** cancella i dati di un Docente dal DB.





5. Glossario

In questa sezione descriveremo i termini tecnici che sono stati utilizzati all'interno del Documento stesso.

- ❖ **API:** Application Program Interface, acronimo che identifica delle librerie che forniscono funzioni implementate.
- ❖ **Form:** insieme di campi da compilare e sottomettere. I campi possono essere obbligatori o facoltativi.
- ❖ **Greenfield:** aggettivo di un progetto che indica la realizzazione dello stesso a partire solamente dall'idea, senza l'utilizzo di componenti già implementate da altri e disponibili online come risorse pubbliche.
- ❖ **Layer:** indica uno strato, un livello, una partizione.
- ❖ **Login:** operazione di autenticazione di un Docente o di un Direttore di Dipartimento mediante l'uso di credenziali personali.
- ❖ **Logout:** disconnessione dalla propria area personale.
- ❖ **MySQL:** è un RDBMS basato sul linguaggio SQL, composto da un client a riga di comando e un server.
- ❖ **Scansione:** lettura di un QR code tramite appropriato dispositivo.
- ❖ **Three Tier:** letteralmente “tre livelli”, è il nome proprio di un pattern utilizzato per la realizzazione dell'architettura software di un Sistema.
- ❖ **Throughput:** quantità di lavoro effettuata in un dato lasso di tempo.
- ❖ **User:** persona che sfrutta le funzionalità del Sistema.
- ❖ **Web Application:** applicazione accessibile via Web per mezzo di una rete Internet.
- ❖ **Web Browser/Browser:** applicazione software per l'acquisizione, la presentazione e la navigazione di risorse sul web.
- ❖ **Web Server:** applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web di un client, tipicamente un Web Browser.