Deliverable D2

Deadline 27/04/2025

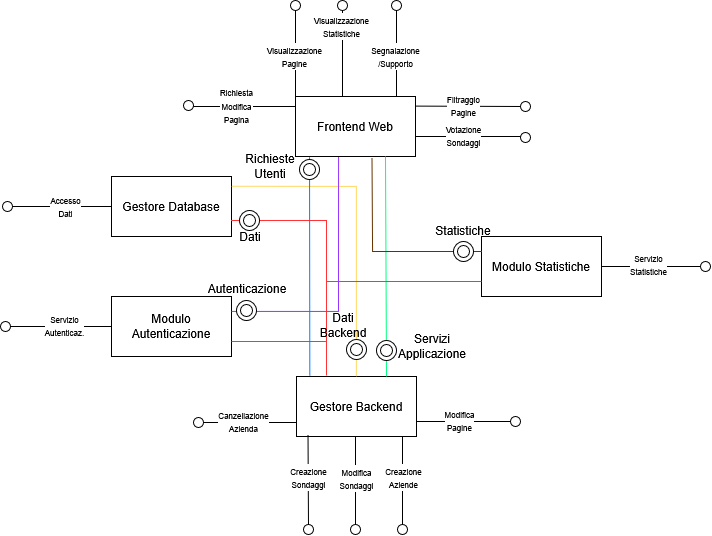
/\* Indicare

• il numero del gruppo

• nome, cognome e numero di matricola di ogni componente del gruppo \*/

1. Diagramma componenti

1.1 Diagramma dei componenti



1.2 Descrizione dei componenti

Gestore Database

Questo componente permette agli altri componenti di accedere e immagazzinare i dati nel database.

| **Tipologia** | **Nome** | **Descrizione** |
| --- | --- | --- |
| Richiesta | Accesso Dati | Il componente utilizza un’interfaccia fornita dal database  per accedere ai dati salvati sul database. |
| Richiesta | Dati Backend | Il componente permette di salvare i dati provenienti dal backend nel database |
| Fornita | Dati | Il componente fornisce un’interfaccia per accedere ai dati  salvati sul database. |

Modulo Autenticazione

Questo componente gestisce l’autenticazione delle aziende

| **Tipologia** | **Nome** | **Descrizione** |
| --- | --- | --- |
| Richiesta | Servizio Autenticazione | Il componente utilizza il servizio di autenticazione per autenticare le aziende |
| Richiesta | Dati | Il componente richiede i dati dal database per verificare l’autenticazione |
| Fornita | Autenticazione | Il componente fornisce l’esito dell’autenticazione |

Modulo Statistiche

Questo componente fornisce le statistiche riguardanti l’utilizzo dell’applicazione

| **Tipologia** | **Nome** | **Descrizione** |
| --- | --- | --- |
| Richiesta | Servizio Statistiche | Il componente utilizza il servizio statistiche per dare il servizio desiderato |
| Richiesta | Dati | Il componente utilizza i dati forniti dal database per poter creare le statistiche |
| Fornita | Statistiche | Il componente fornisce le statistiche |

Gestore Backend

Questo componente gestisce la parte backend dell’applicazione

| **Tipologia** | **Nome** | **Descrizione** |
| --- | --- | --- |
| Richiesta | Richieste Utenti | Il componente riceve le richieste dei vari utenti per fornire i servizi desiderati |
| Richiesta | Dati | Il componente accede ai dati dal database per eseguire le richieste |
| Fornita | Dati Backend | Il componente fornisce i propri dati al database |
| Fornita | Servizi Applicazione | Il componente fornisce i servizi richiesti al frontend |
| Fornita | Cancellazione Azienda | Il componente elimina l’account dell’azienda |
| Fornita | Creazione Sondaggi | Il componente permette la creazione dei sondaggi |
| Fornita | Modifica Sondaggi | Il componente permette la modifica dei sondaggi |
| Fornita | Creazione Azienda | Il componente permette la creazione dell’azienda |
| Fornita | Modifica Pagine | Il componente permette la modifica delle pagine |

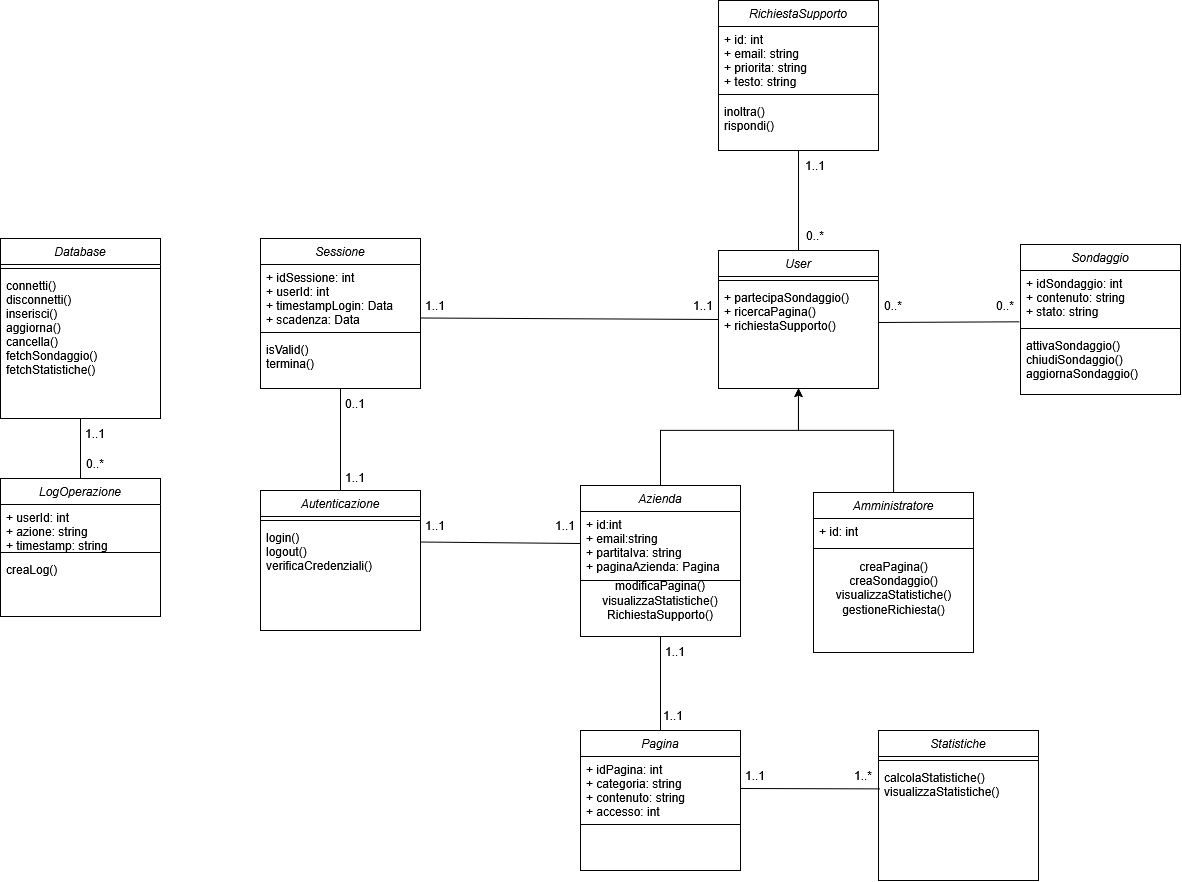
Frontend Web

Questo componente gestisce la parte frontend dell’applicazione

| **Tipologia** | **Nome** | **Descrizione** |
| --- | --- | --- |
| Richiesta | Servizi Applicazioni | Il componente richiede al backend il servizio richiesto |
| Richiesta | Autenticazione | Il componente utilizza l’autenticazione fornita dal modulo autenticazione |
| Richiesta | Statistiche | Il componente richiede le statistiche |
| Fornita | Richieste Utenti | Il componente invia al backend le richieste fornite dai vari utenti |
| Fornita | Richieste Modifica Pagina | Il componente invia al backend la richiesta di modifica della pagina |
| Fornita | Visualizzazione Pagine | Il componente fornisce un’interfaccia per poter visualizzare le varie pagine, anche in base al proprio livello di autenticazione(user, azienda, comune) |
| Fornita | Visualizzazione Statistiche | Il componente fornisce la visualizzazione delle statistiche in base all’autenticazione |
| Fornita | Segnalazione/Supporto | Il componente fornisce un’interfaccia per richiedere supporto per la propria pagina |
| Fornita | Filtraggio Pagina | Il componente fornisce un’interfaccia per filtrare le varie opzioni fornite |
| Fornita | Votazione Sondaggi | Il componente fornisce un’interfaccia per votare ai sondaggi |

2. Diagramma delle classi e descrizione delle classi

2.1 Diagramma delle classi



2.2 Gruppo classi autenticazione e profilo

Vengono analizzate qui le classi che si occupano della gestione dell’autenticazione e del profilo

utente.

2.2.1 User

Questa classe descrive i comportamenti e funzionalità di un utente base senza privilegi

| **Operazione** | **Descrizione** |
| --- | --- |
| partecipaSondaggio() | Invia l’opzione del sondaggio selezionata al database. |
| ricercaPagina() | Ritorna le pagine cercate con i filtri applicati. |
| richiestaSupporto() | Invia una richiesta di supporto all’amministratore. |

2.2.2 Azienda

Questa classe descrive i comportamenti e funzionalità di un’azienda aggiuntive rispetto all’utente

| **Operazione** | **Descrizione** |
| --- | --- |
| modificaPagina() | Invia una richiesta di modifica della pagina all’amministratore, la quale dovrà decidere se approvare o rifiutare la richiesta. |
| visualizzaStatistiche() | Permette di visualizzare le statistiche relative alla pagina aziendale. |
| richiestaSupporto() | Invia una richiesta di supporto all’amministratore con priorità più alta. |

2.2.3 Amministratore

Questa classe descrive i comportamenti e funzionalità dell’amministratore, con i privilegi più alti rispetto alle altre classi

| **Operazione** | **Descrizione** |
| --- | --- |
| creaPagina() | Permette la creazione di una pagina. |
| creaSondaggio() | Permette la creazione di un sondaggio. |
| visualizzaStatistiche() | Permette di visualizzare le statistiche relative al sito. |
| gestioneRichiesta() | Approva o rifiuta la richiesta di modifica di una pagina aziendale. |

2.2.4 Statistiche

Questa classe come vengono elaborate e visualizzate le statistiche dell’applicazione

| **Operazione** | **Descrizione** |
| --- | --- |
| calcolaStatistiche() | Permette di accedere al database per poter ricavare le statistiche ed elaborarle. |
| visualizzaStatistiche() | Permette di visualizzare le statistiche desiderate. |

2.2.5 Sessione

Questa classe gestisce la sessione di utenti autenticati e non

| **Operazione** | **Descrizione** |
| --- | --- |
| isValid() | Ritorna se la sessione desiderata è valida o no. |
| termina() | Termina la sessione scrivendo anche sui log l’azione. |

2.2.5 Autenticazione

Questa classe gestisce l’autenticazione delle aziende

| **Operazione** | **Descrizione** |
| --- | --- |
| login() | Effettua il login confermandola come azienda autenticata |
| logout() | Termina la sessione corrente |
| verificaCredenziali() | Verifica se le credenziali siano corrette |

2.2.6 Richiesta Supporto

Questa classe gestisce le richieste di supporto da parte di utenti e aziende

| **Operazione** | **Descrizione** |
| --- | --- |
| inoltra() | Invia la richiesta all’amministratore |
| rispondi() | L’amministratore può decidere di rispondere alla richiesta |

2.2.7 Sondaggio

Questa classe gestisce i sondaggi creati dall’amministratore

| **Operazione** | **Descrizione** |
| --- | --- |
| attivaSondaggio() | Rende visibile e attivo a tutti il sondaggio selezionato |
| chiudiSondaggio() | Disattiva il sondaggio selezionato |
| aggiornaSondaggio() | Permette la modifica del sondaggio |

2.2.7 Log Operazione

Questa classe gestisce i log delle operazioni effettuate

| **Operazione** | **Descrizione** |
| --- | --- |
| creaLog() | Crea il log inviandolo nel database |

2.2.8 Database

Questa classe gestisce le interazioni con il database

| **Operazione** | **Descrizione** |
| --- | --- |
| connetti() | Permette la connessione al database |
| disconnetti() | Effettua la disconnessione dal database |
| inserisci() | Permette l’uso di “INSERT” nel database |
| aggiorna() | Permette l’uso di “UPDATE” |
| cancella() | Permette l’uso di “DELETE” |
| fetchSondaggio() | Acquisisce i dati riguardanti al sondaggio |
| fetchStatistiche() | Acquisisce i dati riguardanti le statistiche |

3. Constraint OCL

3.1 Autenticazione

| Dopo aver effettuato il login, l’attributo 'autenticato' viene settato a true e gli attributi 'username' e 'password' vengono aggiornati. |
| --- |
| context Autenticazione::login(username: String, password: String): void  post: self.autenticato = true and self.username = username and self.password = password |

| Dopo aver eseguito il logout, l'attributo 'autenticato' viene settato a false |
| --- |
| context Autenticazione::logout(): void  post: self.autenticato = false |

| Verifica che le credenziali inserite siano corrette. |
| --- |
| context Autenticazione::verificaCredenziali(username: String, password: String): Boolean  post: result = (self.username = username and self.password = password) |

| Un oggetto Autenticazione deve avere sia username che password definiti |
| --- |
| context Autenticazione  inv: not self.username.oclIsUndefined() and not self.password.oclIsUndefined() |

3.2 User

| L'utente può partecipare solo a sondaggi attivi. |
| --- |
| context User::partecipaSondaggio(sondaggio: Sondaggio, opzione: String): void  pre: sondaggio.attivo = true  post: sondaggio.voti->includes(opzione) |

| L'utente può inviare una richiesta di supporto con priorità normale. |
| --- |
| context User::richiestaSupporto(): RichiestaSupporto  post: RichiestaSupporto.allInstances()->exists(r | r.richiedente = self and r.priorita = 'normale') |

| Un utente non può essere autenticato senza una sessione valida |
| --- |
| context User  inv: self.autenticato implies self.sessione.isValid() |

3.3 Azienda

| L'azienda può inviare richieste di modifica solo per le pagine di cui è proprietaria. |
| --- |
| context Azienda::modificaPagina(pagina: Pagina): void  pre: pagina.proprietario = self  post: pagina.richiesteModifica->includes(self) |

| L'azienda può inviare richieste di supporto con priorità alta. |
| --- |
| context Azienda::richiestaSupporto(): RichiestaSupporto  post: RichiestaSupporto.allInstances()->exists(r | r.richiedente = self and r.priorita = 'alta') |

| Un'azienda ha una sola pagina collegata |
| --- |
| context Azienda  inv: self.pagine->size() = 1 |

3.4 Amministratore

| L'amministratore può creare nuovi sondaggi. |
| --- |
| context Amministratore::creaSondaggio(titolo: String): Sondaggio  post: Sondaggio.allInstances()->exists(s | s.titolo = titolo and s.creatore = self) |

| L'amministratore può approvare o rifiutare richieste di modifica. |
| --- |
| context Amministratore::gestioneRichiesta(richiesta: RichiestaModifica): void  pre: richiesta.stato = 'in attesa'  post: richiesta.stato = 'approvata' or richiesta.stato = 'rifiutata' |

3.5 Statistiche

| Calcolo delle statistiche basato sui dati del sistema. |
| --- |
| context Statistiche::calcolaStatistiche(): void  post: self.datiCalcolati = true |

3.6 Sessione

| Controlla se la sessione è ancora valida. |
| --- |
| context Sessione::isValid(): Boolean  post: result = (self.dataScadenza > DateTime.now) |

| Termina la sessione e registra l’azione nel log. |
| --- |
| context Sessione::termina(): void  post: self.attiva = false and LogOperazione.allInstances()->exists(l | l.azione = 'logout' and l.utente = self.utente) |

| Se la sessione è attiva, deve avere una data di scadenza futura |
| --- |
| context Sessione  inv: self.attiva implies self.dataScadenza > DateTime.now |

3.7 Richiesta Supporto

| Risponde alla richiesta e la completa. |
| --- |
| context RichiestaSupporto::rispondi(risposta: String): void  pre: self.stato = 'inoltrata'  post: self.risposta = risposta and self.stato = 'completata' |

3.8 Sondaggio

| Attiva un sondaggio, rendendolo visibile a tutti. |
| --- |
| context Sondaggio::attivaSondaggio(): void  post: self.attivo = true |

| Chiude un sondaggio, rendendolo inattivo. |
| --- |
| context Sondaggio::chiudiSondaggio(): void  post: self.attivo = false |

| Un sondaggio deve avere almeno 2 opzioni per essere valido |
| --- |
| context Sondaggio  inv: self.opzioni->size() >= 2 |

3.9 Log Operazione

| Crea un log per una determinata azione nel sistema. |
| --- |
| context LogOperazione::creaLog(azione: String): LogOperazione  post: LogOperazione.allInstances()->exists(l | l.azione = azione and l.timestamp = DateTime.now) |

| Ogni log deve avere una azione e un timestamp |
| --- |
| context LogOperazione  inv: not self.azione.oclIsUndefined() and not self.timestamp.oclIsUndefined() |

3.10 Database

| Attiva la connessione al database. |
| --- |
| context Database::connetti(): void  post: self.connessioneAttiva = true |

| Disattiva la connessione al database. |
| --- |
| context Database::disconnetti(): void  post: self.connessioneAttiva = false |

| Inserisce un nuovo dato se la connessione è attiva. |
| --- |
| context Database::inserisci(dato: String): void  pre: self.connessioneAttiva = true  post: self.dati->includes(dato) |

| Aggiorna un dato esistente nel database |
| --- |
| context Database::aggiorna(datoVecchio: String, datoNuovo: String): void  pre: self.dati->includes(datoVecchio)  post: not self.dati->includes(datoVecchio) and self.dati->includes(datoNuovo) |

| Cancella un dato esistente dal database. |
| --- |
| context Database::cancella(dato: String): void  pre: self.dati->includes(dato)  post: not self.dati->includes(dato) |