

Progetto Basi di dati I

Alessio Mobilia, Marco Petix

SmartCampus

Il progetto si articola intorno all'inserimento di un campus universitario all'interno di un contesto "smart".

La piattaforma punta quindi a gestire ed informatizzare l'interazione del personale universitario e del corpo studentesco con i locali e le infrastrutture presenti all'interno del plesso universitario.

Utenza

I ruoli che caratterizzano l'utenza ricalcano le figure presenti nel contesto universitario. Ipotizziamo che i dati anagrafici degli utenti, così come le credenziali di accesso alla piattaforma, siano memorizzati insieme ad altre informazioni specifiche, all'interno di una banca dati dedicata.

Esaminiamo i vari tipi di utente previsti:

- il **DOCENTE** appartiene ad un determinato **dipartimento** ed usufruisce di un **ufficio** all'interno del campus. Tiene anche delle **lezioni** in determinati locali ad orari definiti.
- lo **STUDENTE** è associato ad una **matricola** e ad un determinato **corso di laurea** da cui dipende l'orario delle **lezioni** da seguire.
- Il **PERSONALE AUSILIARIO**, come l'addetto alla sicurezza e l'addetto alle pulizie ha accesso a determinati settori del campus in relazione ai **turni** previsti dal loro contratto.
- il **PERSONALE AMMINISTRATIVO** lavora in un determinato **locale** e garantisce un servizio agli studenti ed al resto del personale secondo dei **turni** definiti.

Edifici

Il campus è composto da diversi plessi. Ogni plesso è dedicato ad uno o più dipartimenti e racchiude i vari locali.

La posizione fisica di un **LOCALE** appartenente al campus, è definita dal **plesso** di appartenenza, dal **piano** e dal **locale** che lo contiene (es. corridoio). Altre informazioni specifiche sono associate ai vari locali in base alla tipologia degli stessi.

Esaminiamo i vari locali presenti all'interno di un'ipotetica università:

- le **AULE** ospitano le **lezioni** tenute dai **docenti**.
- gli **UFFICI** sono caratterizzati dai **docenti** che ne usufruiscono e dall'**orario di ricevimento** definito dagli stessi.
- I **LOCALI SPECIALI** sono gestiti dal **personale amministrativo** e caratterizzati da un **orario di attività**, come ad esempio biblioteche, uffici amministrativi e laboratori.

L'uscita dal campus si concretizza con l'interazione con il locale "**Fuori dal campus**" di tipo "**Altri locali**".

La **look-up table** relativa ai **tipi di locale** permette l'implementazione di nuove tipologie di strutture.

Dispositivi

Ipotizziamo che i locali del campus siano dotati di vari dispositivi:

- ogni utente è munito di un **dispositivo d'identificazione** come una smart-card o di una applicazione che interagisca con dei **dispositivi d'accesso** (es. QR-code) associati ai vari locali. Questi sono poi utilizzati per interagire con il sistema e registrare l'entrata e l'uscita di ogni utente.
- ogni accesso è quindi dotato di lettori smart-card o simili che permettono di interagire con l'utenza. Ai suddetti lettori si accosta un monitor che permetta agli utenti di ricevere un **feedback** riguardo l'**esito** della loro **richiesta di accesso**.
- i corridoi e gli altri locali del campus sono muniti di **dispositivi audio e video** (es. monitor e speaker) in grado di riprodurre in streaming contenuti audio e video.
- gli impianti di **illuminazione e riscaldamento** sono in grado di essere manipolati tramite apposite interfacce e oltre che dalla eventuale presenza di utenti nei locali del campus.

Servizi

Accesso ai locali

L'obiettivo è costruire uno storico degli spostamenti e della presenza di ogni utente all'interno del perimetro del campus.

Esaminiamo l'interazione tra l'utenza ed i vari locali presenti all'interno di un'ipotetica università:

- le **AULE** sono chiuse agli **studenti** se non in corrispondenza delle **lezioni**, evidenziate nei vari orari. È comunque necessaria la presenza di un **docente** che sia presente in aula al momento dell'accesso degli studenti. Due o più lezioni non possono condividere contemporaneamente la stessa aula.

- ogni **docente** è dotato di un **UFFICIO**, identificato anch'esso dal **piano** e dal **dipartimento** che lo ospita. L'accesso è sempre garantito al proprietario e quest'ultimo può definire un **orario di ricevimento** che garantirà l'accesso ad altri utenti, sempre a patto che il professore sia presente.
- i **LOCALI SPECIALI** sono accessibili all'utenza ad **orari definiti** dal personale amministrativo.
- gli addetti alla **sicurezza** ed alla **pulizia** hanno accesso a tutti i locali presenti nel plesso e nel piano definito dal proprio **turno** di lavoro. Gli addetti alla pulizia non potranno accedere alle aule mentre sono utilizzate.

Interazione con i dispositivi

Oltre alla gestione degli accessi il progetto include l'automatizzazione dell'illuminazione del campus.

L'illuminazione all'interno delle aule verrà attivata all'accesso del primo utente e rimarrà attiva fino allo svuotamento dell'aula stessa.

E' possibile selezionare tutti i dispositivi presenti in un determinato locale oppure tutti quelli presenti nel campus e successivamente attivarli o disattivarli secondo le necessità. Questa capacità spetterà solamente al personale con poteri da amministratore.

Documentazione del diagramma E-R

Entità	Descrizione	Attributi	Identificatore
Utente	Generalizzazione delle persone	Nome, Cognome, Username, Password, Data di nascita, ID_Utente	ID
Studente	Specializzazione di utente. Partecipa alle lezioni ed è iscritto ad un cdl	Nome, Cognome, Data di nascita, Username, Password, ID_Utente, Matricola	Matricola
Tirocinante	Specializzazione di studente.	Nome, Cognome, Data di nascita, Username, Password,	Matricola

		ID_Utente, Matricola	
Docente	Specializzazione di utente. Insegna una o più materie.	Nome, Cognome, Data di nascita, Username, Password, ID_Utente, idDocente	idDocente
Personale	Specializzazione di utente che si occupa di mansioni predefinite.	Nome, Cognome, Data di nascita, Username, Password, ID_Utente, idPersonale	idPersonale
Personale Ausiliario	Specializzazione di Personale addetto alla sicurezza o alla pulizia dei locali.	Nome, Cognome, Data di nascita, ID_Utente, idPersonale, Username, Password, Sicurezza	idPersonale
Personale amministrativo	Specializzazione di Personale che svolge mansioni di amministrazione all'interno di un ufficio	Nome, Cognome, Data di nascita, ID_Utente, idPersonale, Username, Password, Amministratore	idPersonale
Dispositivo di identificazione	Dispositivo che identifica univocamente una persona.	ID	ID
Dispositivi Locali	Un dispositivo audio/video, di illuminazione o di accesso presente all'interno di un locale	ID, Attivo, nome, MAC_address	ID
Tipo dispositivi locale	look-up table che contiene i possibili tipi di dispositivi dei locali	Nome	Nome
Corso di laurea	Corsi di studio offerti dall'ateneo	Nome	Nome

Dipartimento	Entità che contiene i dipartimenti previsti dall'ateneo	Nome	Nome
Materia	Materia facente parte di un determinato cdl	Nome	Nome
Turni Ricevimento	Intervalli di tempo dei docenti riservati al colloquio con gli studenti	Inizio, fine	Inizio, fine, ID_Docente
Lezione	Entità che raggruppa gli orari in cui sono tenute le lezioni	Inizio, Fine	Inizio, Fine, Nome Materia.
Locale	Generalizzazione (non totale) delle aree interne agli edifici universitari. Contiene gli ambienti universitari sia corridoi che stanze.	ID, nome, piano, Disponibilità	ID
Locale Speciale	Specializzazione di Locale. Contiene tutti i locali il cui accesso è possibile solo a determinati orari.	ID, nome, piano, Disponibilità	ID
Aula	Specializzazione di Stanza. Ospita le lezioni	ID, nome, piano, Disponibilità	ID
Ufficio	Specializzazione di stanza. Contiene gli uffici dei docenti.	ID, nome, piano, Disponibilità	ID
Calendario	Contiene gli orari a cui i locali speciali sono accessibili agli studenti	Inizio, Fine	Inizio, Fine, ID_locale
Plesso	Contiene l'elenco dei plessi del campus.	nome, numero di piani.	nome
Log	Conserva lo storico	Timestamp	Timestamp,

	degli accessi degli utenti ai locali		ID_Utente, ID_Locale
Tipo Esito	Look-up table che contiene i possibili tipi di esito dei tentativi di accesso ai locali conservati nel log	codice, descrizione	codice
Presenza	E' presente una entry per ogni utente attualmente presente nel campus e conserva il luogo in cui si trova.	ID	ID
Turni del Personale Amministrativo	Intervalli di tempo che definiscono le mansioni degli impiegati amministrativi	Inizio, Fine	Ora Inizio, Ora Fine, Data, ID_Utente
Turni del Personale Ausiliario	Intervalli di tempo e luoghi che definiscono le mansioni degli impiegati ausiliari	Inizio, Fine, Piano	Inizio, Fine, Piano, ID_Utente

Relazione	Descrizione	Entità coinvolte	Attributi
Associazione	Associa un Utente ai propri dispositivi di identificazione.	Persona (1,N) Dispositivo di identificazione (1,1)	
Posizione	Associa i dati di log al Locale	Locale(0,N) Log(1,1)	
Soggetto	Associa i dati di log all'utente	Utente(0,N) Log(1,1)	
Iscrizione	Associa uno studente al corso di laurea a cui è iscritto.	Studente (1,1) Corso di laurea (1,N)	
Partecipazione	Associa un docente al proprio	Docente (1,1) Dipartimento(1,N)	

	dipartimento.		
Contiene	Associa ogni corso di laurea al proprio dipartimento.	Corso di Laurea (1,1) Dipartimento(1,N)	
Ricevimento	Associa il docente ai propri orari di ricevimento nella tabella Turni.	Docente (1,N) Turni (1,1)	
Occupava	Associa la Lezione con l'aula in cui deve essere svolta	Lezione(1,1) Aula(0,N)	
Appartenere	Associa il Dipartimento al Locale dove è sito	Dipartimento (0,N), Locale (0,1)	
Manifesto	Associa il corso di laurea con le materie previste dai relativi manifesti	Corso di Laurea(1,N) Materia(1,1)	
Docenza	Associa il Docente alle materie che insegna	Docente(0,N) Materia(1,1)	
Insegnare	Associa la lezione con il docente che la deve svolgere	Lezione (1,1) Docente(0,N)	
Argomento	Associa La lezione con la relativa materia	Lezione(1,1) Materia(1,N)	
Fruizione	Associa un docente con il proprio ufficio.	Docente (0,1) Ufficio(1,N)	
Contratto Amministrativo	Associa il personale amministrativo con i turni previsti dal proprio contratto	Personale Amministrativo(0,N), Turno del Personale Amministrativo (1,1)	
Contratto Ausiliario	Associa il personale ausiliario con i turni previsti dal proprio contratto	Personale(0,N), Turno del Personale Ausiliario(1,1)	
Sede Contratto Amministrativo	Associa i turni previsti da un	Turno del Personale Amministrativo (1,1),	

	contratto per dipendenti amministrativi con il locale in cui si svolge	Locale (0,N)	
Include	Associa dei locali con il locali in esso contenuti	Locale(0,N) Locale(0,N)	
Dotazione	Associa i dispositivi con i locale in cui si trovano	Dispositivo Locale(1,1) Locale(1,N)	
Accesso	Associa i locali speciali ai turni in cui è accessibile.	Locale Speciale(1,N) Calendario(1,1)	
Tirocinio	Lega i tirocinanti con i locali in cui viene svolto il tirocinio e il Docente responsabile	Tirocinante(1,1) Locale(0,N) Docente(0,N)	descrizione
Associazione tipologia studente	Lega i dispositivi dei locali alla look-up table dei tipi dei dispositivi	Dispositivi Locale (1,1) tipo dispositivi locale (0,N)	
Esito	Lega i log alla look-up table degli esiti degli accessi	Log (1,1) Tipo Esito (0,N)	
Presenza Utente	Associa l'utente alla entry nella tabella presenza	Utente (0,N) Presenza(1,1)	
Associazione di presenza	Associa il locale alla tabella presenza	Locale (0,N) Presenza(1,1)	

Regole di Vincolo	
GENERALI	
RV1	Tutti gli Utenti possono accedere ai locali speciali nelle fasce orarie indicate nel "calendario"
RV2	I non addetti alle pulizie non possono entrare in una stanza dove si trova il personale delle pulizie.

RV3	Tutti possono accedere sempre ai locali di tipo “corridoio” e “altri locali” a meno che non siano classificati come “non disponibili”.
RV4	Nessuno può accedere a locali “non disponibili”.
RV5	I dispositivi di tipo “luce” di un determinato locale verranno attivati non appena un utente effettuerà un accesso allo stesso.
RV6	I dispositivi di tipo “luce” di un determinato locale verranno disattivati all’uscita dell’ultimo utente presente all’interno dello stesso.
RV7	Non vi possono essere due lezioni in contemporanea nella stessa aula.
STUDENTI	
RV8	Uno studente può accedere all’ufficio del docente solo nelle fasce orarie indicate nell’entità “Turni ricevimento” e solamente se il Docente è presente all’interno dell’ufficio stesso.
RV9	Uno studente può accedere alle aule solamente se è prevista una lezione a quell’ora e se il docente è presente all’interno dell’aula.
RV10	Gli studenti possono sempre entrare nel locale dove svolgono il tirocinio.
DOCENTI	
RV11	Il docente può entrare in aula solamente se vi è una lezione.
RV12	Il docente può sempre entrare nel proprio ufficio.
RV13	Il docente può entrare nell’ufficio di altri docenti solo se questi sono presenti nel locale
PERSONALE	
RV14	Solo il Personale con privilegi di amministratore può visionare il Log relativo a gli spostamenti di un determinato utente o alle statistiche di accesso ai locali.
RV15	Il personale della sicurezza può entrare in qualsiasi locale secondo gli orari previsti dal proprio turno di lavoro.
RV16	Il personale delle pulizie può accedere ad un locale solo se non vi sono altri tipi di utenti all’interno del locale.
RV17	Il personale amministrativo può accedere nel locale indicato dal suo turno durante gli orari previsti dal turno stesso indipendentemente dall’orario di apertura del locale in questione.
RV18	Il personale amministrativo può accedere alle aule solamente se è prevista una lezione a quell’ora e se il docente è nell’aula.
RV19	il personale amministrativo può entrare nell’ufficio del docente se questo è nel suo ufficio.

Regole di Derivazione	
RD1	Ogni volta che un soggetto accede ad un locale vengono conservate le informazioni in log.
RD2	Ogni volta che un utente accede con successo ad un locale viene aggiornata la tabella presenza, in questa è indicato il luogo dove si trova attualmente l'utente.
RD3	Quando un utente esce dal campus la relativa entry nella tabella presenze viene eliminata.
RD4	Il campo "piano" (in locale e in personale) non può essere maggiore del numero piani del plesso corrispondente.
RD5	Il campo esito nella tabella log contiene il codice relativo all'esito del tentativo di accesso al locale.
RD6	L'ufficio del docente deve far parte dello stesso Dipartimento del Docente.
RD7	Per ogni entry della tabella turno personale se esiste il campo plesso non può esistere il campo locale e viceversa.
RD8	L'idLocale nella tabella ricevimento è uguale all'id dell'ufficio del professore che fa il ricevimento.

Diagramma concettuale

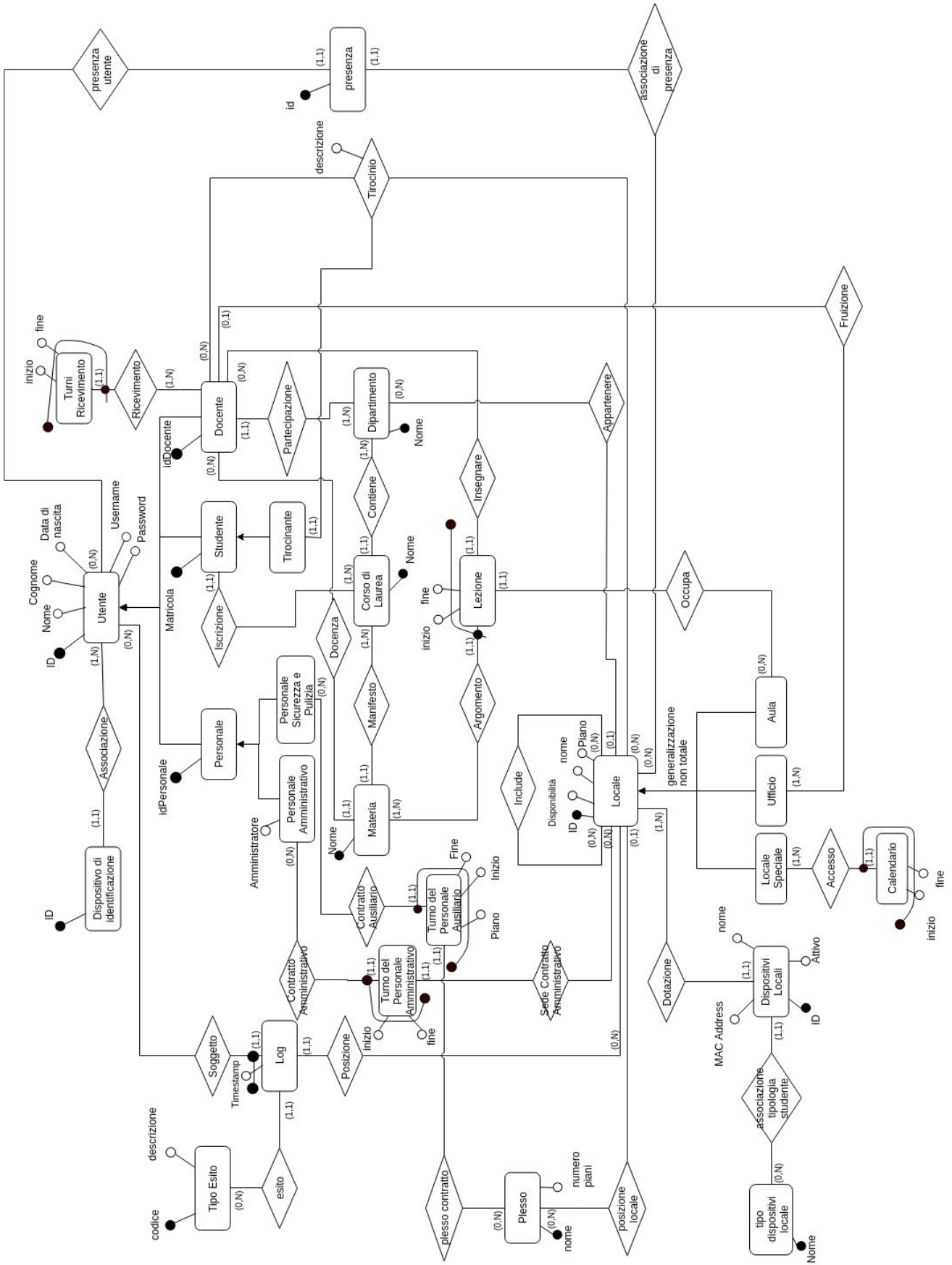


Diagramma concettuale ottimizzato

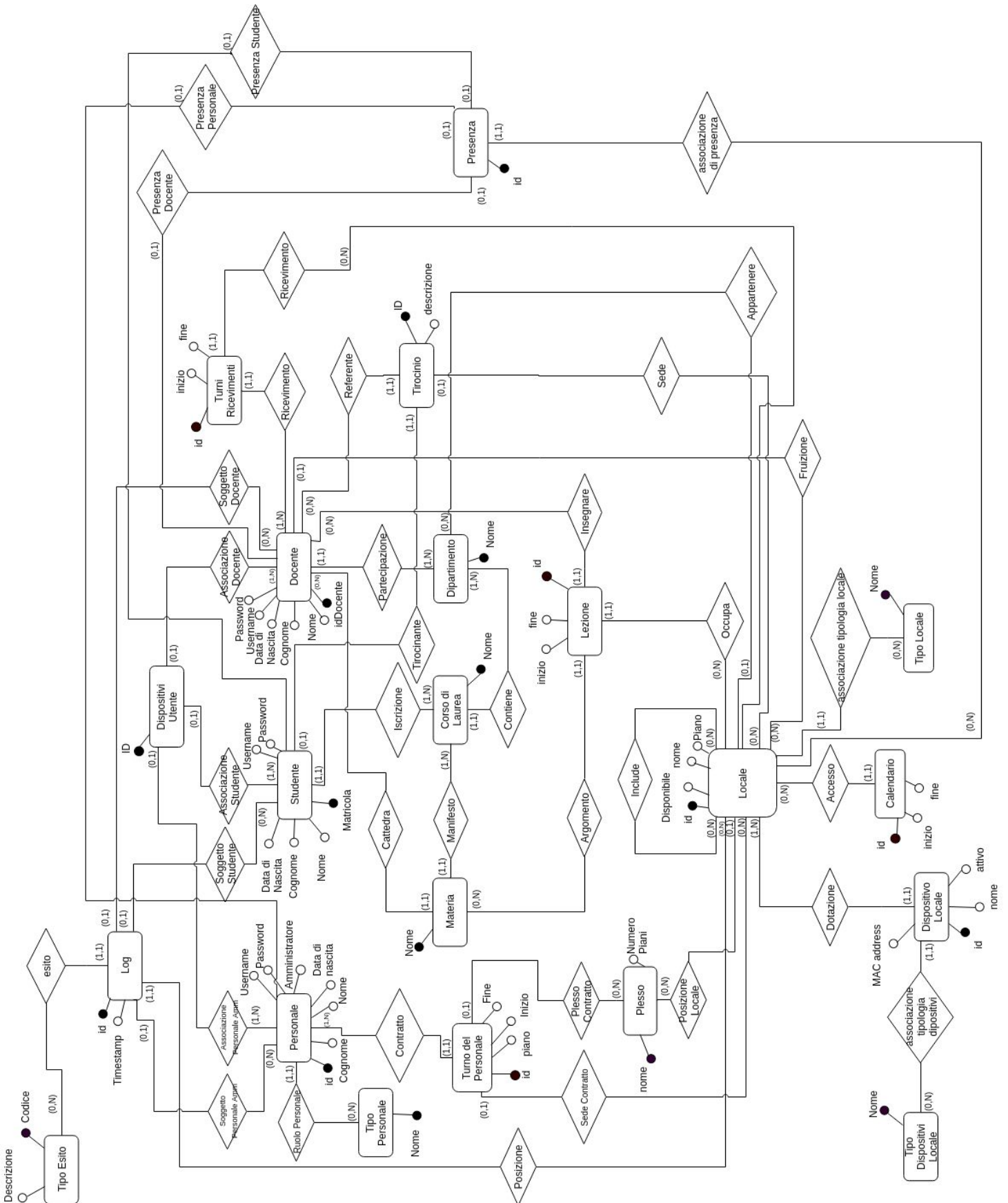


Diagramma Logico

calendario (**id**, inizio, fine, *idLocale*),
corso_di_laurea (**nome**, *dipartimento*),
dipartimento (**nome**),
dispositivi_utente (**id**, *idStudente*, *idDocente*, *idPersonale*),
dispositivo_locale (**id**, *MAC_address*, nome, *tipo*, attivo, *idLocale*),
docente (**idDocente**, cognome, nome, dataNascita, *dipartimento*, *idUfficio*, username, **password**),
lezione (**id**, inizio, fine, *materia*, *idLocale*, *idDocente*),
locale (**id**, nome, tipo, disponibile, *plesso*, piano, *dipartimento*, *localeContenitore*),
log (**id**, timestamp, esito, *idStudente*, *idDocente*, *idPersonale*, *idLocale*),
materia (**nome**, *corsoDiLaurea*, *idDocente*),
personale (**idPersonale**, cognome, nome, dataNascita, *tipo*, amministratore, **username**, **password**),
plesso (**nome**, numeroPiani),
presenza (**id**, *idStudente*, *idDocente*, *idPersonale*, *idLocale*),
studente (**matricola**, cognome, nome, dataNascita, *corsoDiLaurea*, username, **password**),
tipo_dispositivi_locale (**nome**),
tipo_esito (**codice**, descrizione),
tipo_locale (**nome**),
tipo_personale (**nome**),
 tirocinio (**id**, descrizione, *idLocale*, *idStudente*, *idDocente*),
turni_personale (**id**, inizio, fine, *plesso*, piano, *idLocale*, *idPersonale*),
turni_ricevimenti (**id**, inizio, fine, *idDocente*, *idLocale*).

entità
attributo generico
chiave primaria
chiave esterna

Database

Calendario

```
CREATE TABLE `calendario` (  
  `id` int(11) NOT NULL,  
  `inizio` datetime NOT NULL,  
  `fine` datetime NOT NULL,  
  `idLocale` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `calendario`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `id_calendario` (`id`),  
  ADD KEY `accesso_calendario_locale` (`idLocale`);
```

```
ALTER TABLE `calendario`  
  ADD CONSTRAINT `accesso_calendario_locale` FOREIGN KEY (`idLocale`)  
  REFERENCES `locale` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Trigger

InsertCalendario [BEFORE INSERT]: annulla ogni insert il cui campo idLocale non faccia riferimento ad un locale di tipo 'locale speciale'.

Corso_di_laurea

```
CREATE TABLE `corso_di_laurea` (  
  `nome` varchar(50) NOT NULL,  
  `dipartimento` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `corso_di_laurea`
```

```
ADD PRIMARY KEY (`nome`),  
ADD KEY `contiene_corso_di_laurea_dipartimento` (`dipartimento`);
```

```
ALTER TABLE `corso_di_laurea`  
ADD CONSTRAINT `contiene_corso_di_laurea_dipartimento` FOREIGN KEY  
(`dipartimento`) REFERENCES `dipartimento` (`nome`) ON UPDATE CASCADE;
```

Dipartimento

```
CREATE TABLE `dipartimento` (  
  `nome` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `dipartimento`  
ADD PRIMARY KEY (`nome`);
```

Dispositivi_utente

```
CREATE TABLE `dispositivi_utente` (  
  `id` int(11) NOT NULL,  
  `MAC` bigint(20) NOT NULL DEFAULT '0',  
  `idStudente` int(11) DEFAULT NULL,  
  `idDocente` int(11) DEFAULT NULL,  
  `idPersonale` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `dispositivi_utente`  
ADD PRIMARY KEY (`id`),  
ADD KEY `id_dispositivi_utente` (`id`),  
ADD KEY `associazione_dispositivi_utente_docente` (`idDocente`),  
ADD KEY `associazione_dispositivi_utente_pamministrativo`  
(`idPersonale`),  
ADD KEY `associazione_dispositivi_utente_studente` (`idStudente`);
```

```

ALTER TABLE `dispositivi_utente`
  ADD CONSTRAINT `associazione_dispositivi_utente_docente` FOREIGN KEY
  (`idDocente`) REFERENCES `docente` (`idDocente`) ON DELETE NO ACTION ON
  UPDATE CASCADE,
  ADD CONSTRAINT `associazione_dispositivi_utente_studente` FOREIGN KEY
  (`idStudente`) REFERENCES `studente` (`matricola`) ON DELETE NO ACTION
  ON UPDATE CASCADE,
  ADD CONSTRAINT `associazione_dispositivi_utenti_personale` FOREIGN KEY
  (`idPersonale`) REFERENCES `personale` (`idPersonale`) ON DELETE NO
  ACTION ON UPDATE CASCADE;

```

Trigger

Insert_disp_utente [BEFORE INSERT]: si assicura che per ogni insert solo uno tra idStudente, idDocente ed idPersonale sia diverso da NULL.

Dispositivo_locale

```

CREATE TABLE `dispositivo_locale` (
  `id` int(11) NOT NULL,
  `MAC_Address` bigint(20) DEFAULT NULL,
  `nome` varchar(50) DEFAULT NULL,
  `tipo` varchar(50) NOT NULL,
  `attivo` tinyint(1) NOT NULL DEFAULT '0',
  `idLocale` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

ALTER TABLE `dispositivo_locale`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id_dispositivo_locale` (`id`),
  ADD KEY `dotazione_dispositivi_locale_locale` (`idLocale`),
  ADD KEY `ruolo_dispositivi_locale_tipo` (`tipo`);

```

```

ALTER TABLE `dispositivo_locale`
  ADD CONSTRAINT `dotazione_dispositivi_locale_locale` FOREIGN KEY
  (`idLocale`) REFERENCES `locale` (`id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `ruolo_dispositivi_locale_tipo` FOREIGN KEY (`tipo`)
  REFERENCES `tipo_dispositivi_locale` (`nome`) ON UPDATE CASCADE;

```


Docente

```
CREATE TABLE `docente` (  
  `idDocente` int(11) NOT NULL,  
  `cognome` varchar(30) NOT NULL,  
  `nome` varchar(30) NOT NULL,  
  `dataNascita` date NOT NULL,  
  `dipartimento` varchar(50) NOT NULL,  
  `idUfficio` int(11) DEFAULT NULL,  
  `username` varchar(30) NOT NULL,  
  `password` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `docente`  
  ADD PRIMARY KEY (`idDocente`),  
  ADD UNIQUE KEY `username` (`username`),  
  ADD KEY `id_docente` (`idDocente`),  
  ADD KEY `partecipazione_docente_dipartimento` (`dipartimento`),  
  ADD KEY `fruizione_docente_locale` (`idUfficio`);
```

```
ALTER TABLE `docente`  
  ADD CONSTRAINT `fruizione_docente_locale` FOREIGN KEY (`idUfficio`)  
  REFERENCES `locale` (`id`) ON DELETE SET NULL ON UPDATE CASCADE,  
  ADD CONSTRAINT `partecipazione_docente_dipartimento` FOREIGN KEY  
  (`dipartimento`) REFERENCES `dipartimento` (`nome`) ON UPDATE CASCADE;
```

Trigger

InsertDocente [BEFORE INSERT]: annulla ogni insert il cui campo idUfficio non faccia riferimento ad un locale di tipo 'ufficio'.

Lezione

```
CREATE TABLE `lezione` (  
  `id` int(11) NOT NULL,  
  `inizio` datetime NOT NULL,  
  `fine` datetime NOT NULL,  
  `materia` varchar(50) NOT NULL,  
  `idLocale` int(11) NOT NULL,
```

```
`idDocente` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `lezione`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id_lezione` (`id`),
  ADD KEY `argomento_lezione_materia` (`materia`),
  ADD KEY `insegnare_lezione_docente` (`idDocente`),
  ADD KEY `occupa_lezione_locale` (`idLocale`);
```

```
ALTER TABLE `lezione`
  ADD CONSTRAINT `argomento_lezione_materia` FOREIGN KEY (`materia`)
REFERENCES `materia` (`nome`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `insegnare_lezione_docente` FOREIGN KEY (`idDocente`)
REFERENCES `docente` (`idDocente`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `occupa_lezione_locale` FOREIGN KEY (`idLocale`)
REFERENCES `locale` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Trigger

InsertLezione [BEFORE INSERT]: annulla ogni insert il cui campo idLocale non faccia riferimento ad un locale di tipo 'Aula'.

Locale

```
CREATE TABLE `locale` (
  `id` int(11) NOT NULL,
  `nome` varchar(100) DEFAULT NULL,
  `tipo` varchar(50) NOT NULL,
  `disponibile` tinyint(1) NOT NULL DEFAULT '1',
  `plesso` varchar(30) DEFAULT NULL,
  `piano` int(11) NOT NULL,
  `dipartimento` varchar(50) DEFAULT NULL,
  `localeContentitore` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `locale`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id_locale` (`id`),
```

```

ADD KEY `include_locale_locale` (`localeContenitore`),
ADD KEY `appartenere_locale_dipartimento` (`dipartimento`),
ADD KEY `ruolo_locale_tipo` (`tipo`),
ADD KEY `ruolo_locale_plesso` (`plesso`);

```

```

ALTER TABLE `locale`
  ADD CONSTRAINT `appartenere_locale_dipartimento` FOREIGN KEY
  (`dipartimento`) REFERENCES `dipartimento` (`nome`) ON DELETE SET NULL
  ON UPDATE CASCADE,
  ADD CONSTRAINT `include_locale_locale` FOREIGN KEY
  (`localeContenitore`) REFERENCES `locale` (`id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `ruolo_locale_plesso` FOREIGN KEY (`plesso`) REFERENCES
  `plesso` (`nome`) ON UPDATE CASCADE,
  ADD CONSTRAINT `ruolo_locale_tipo` FOREIGN KEY (`tipo`) REFERENCES
  `tipo_locale` (`nome`) ON UPDATE CASCADE;

```

Trigger

InsertLocale [BEFORE INSERT]: annulla ogni insert il cui campo Piano faccia riferimento ad un piano non presente nel plesso indicato dal campo Plesso.

Log

```

CREATE TABLE `log` (
  `id` int(11) NOT NULL,
  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
  CURRENT_TIMESTAMP,
  `esito` int(11) NOT NULL DEFAULT '9999',
  `idStudente` int(11) DEFAULT NULL,
  `idDocente` int(11) DEFAULT NULL,
  `idPersonale` int(11) DEFAULT NULL,
  `idLocale` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

ALTER TABLE `log`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id_log` (`id`),
  ADD KEY `posizione_log_locale` (`idLocale`),
  ADD KEY `soggetto_log_docente` (`idDocente`),
  ADD KEY `soggetto_log_personale_ausiliario` (`idPersonale`),

```

```
ADD KEY `soggetto_log_studente` (`idStudente`),
ADD KEY `esito_log_tipo_esito` (`esito`);
```

```
ADD CONSTRAINT `esito_log_tipo_esito` FOREIGN KEY (`esito`) REFERENCES
`tipo_esito` (`codice`) ON DELETE NO ACTION ON UPDATE CASCADE,
ADD CONSTRAINT `posizione_log_locale` FOREIGN KEY (`idLocale`)
REFERENCES `locale` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD CONSTRAINT `soggetto_log_docente` FOREIGN KEY (`idDocente`)
REFERENCES `docente` (`idDocente`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
ADD CONSTRAINT `soggetto_log_personale` FOREIGN KEY (`idPersonale`)
REFERENCES `personale` (`idPersonale`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
ADD CONSTRAINT `soggetto_log_studente` FOREIGN KEY (`idStudente`)
REFERENCES `studente` (`matricola`) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```

Trigger

InsertLog [BEFORE INSERT]: filtra ogni insert in base alle interazioni utente-locale e genera un esito corrispondente per ogni interazione. In caso di esito positivo, esito uguale a 0, provvede anche ad aggiornare la tabella Presenza.

Materia

```
CREATE TABLE `materia` (
  `nome` varchar(50) NOT NULL,
  `corsoDiLaurea` varchar(50) NOT NULL,
  `idDocente` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `materia`
ADD PRIMARY KEY (`nome`),
ADD KEY `manifesto_materia_corso_di_laurea` (`corsoDiLaurea`),
ADD KEY `docenza_materia_docente` (`idDocente`);
```

```
ALTER TABLE `materia`
ADD CONSTRAINT `docenza_materia_docente` FOREIGN KEY (`idDocente`)
REFERENCES `docente` (`idDocente`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `manifesto_materia_corso_di_laurea` FOREIGN KEY
(`corsoDiLaurea`) REFERENCES `corso_di_laurea` (`nome`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

Personale

```
CREATE TABLE `personale` (
  `idPersonale` int(11) NOT NULL,
  `cognome` varchar(50) NOT NULL,
  `nome` varchar(50) NOT NULL,
  `dataNascita` date NOT NULL,
  `tipo` varchar(30) NOT NULL,
  `amministratore` tinyint(1) NOT NULL,
  `username` varchar(30) NOT NULL,
  `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `personale`
  ADD PRIMARY KEY (`idPersonale`),
  ADD UNIQUE KEY `username` (`username`),
  ADD KEY `id_personale` (`idPersonale`),
  ADD KEY `ruolo_personale_tipo` (`tipo`);
```

```
ALTER TABLE `personale`
  ADD CONSTRAINT `ruolo_personale_tipo` FOREIGN KEY (`tipo`) REFERENCES
`tipo_personale` (`nome`) ON UPDATE CASCADE;
```

Plesso

```
CREATE TABLE `plesso` (
  `nome` varchar(50) NOT NULL,
  `numeroPiani` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `plesso`
  ADD PRIMARY KEY (`nome`);
```

Presenza

```
CREATE TABLE `presenza` (  
  `id` int(11) NOT NULL,  
  `idStudente` int(11) DEFAULT NULL,  
  `idDocente` int(11) DEFAULT NULL,  
  `idPersonale` int(11) DEFAULT NULL,  
  `idLocale` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `presenza`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `id` (`id`),  
  ADD KEY `posizionePresenzaDocente` (`idDocente`),  
  ADD KEY `posizionePresenzaLocale` (`idLocale`),  
  ADD KEY `posizionePresenzaPersonale` (`idPersonale`),  
  ADD KEY `posizionePresenzaStudente` (`idStudente`);
```

```
ALTER TABLE `presenza`  
  ADD CONSTRAINT `posizionePresenzaDocente` FOREIGN KEY (`idDocente`)  
  REFERENCES `docente` (`idDocente`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `posizionePresenzaLocale` FOREIGN KEY (`idLocale`)  
  REFERENCES `locale` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `posizionePresenzaPersonale` FOREIGN KEY  
  (`idPersonale`) REFERENCES `personale` (`idPersonale`) ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  ADD CONSTRAINT `posizionePresenzaStudente` FOREIGN KEY (`idStudente`)  
  REFERENCES `studente` (`matricola`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Trigger

Insert_presenza [BEFORE INSERT]: si assicura che per ogni insert solo uno tra idStudente, idDocente ed idPersonale sia diverso da NULL.

Studente

```
CREATE TABLE `studente` (  
  `matricola` int(11) NOT NULL,  
  `cognome` varchar(30) NOT NULL,  
  `nome` varchar(30) NOT NULL,  
  `dataNascita` date NOT NULL,  
  `corsoDiLaurea` varchar(50) NOT NULL,  
  `username` varchar(30) NOT NULL,  
  `password` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `studente`  
  ADD PRIMARY KEY (`matricola`),  
  ADD UNIQUE KEY `username` (`username`),  
  ADD KEY `matricola_studente` (`matricola`),  
  ADD KEY `iscrizione_studente_corso_di_laurea` (`corsoDiLaurea`);
```

```
ALTER TABLE `studente`  
  ADD CONSTRAINT `iscrizione_studente_corso_di_laurea` FOREIGN KEY  
  (`corsoDiLaurea`) REFERENCES `corso_di_laurea` (`nome`) ON UPDATE  
  CASCADE;
```

Tipo_dispositivi_locale

```
CREATE TABLE `tipo_dispositivi_locale` (  
  `nome` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tipo_dispositivi_locale`  
  ADD PRIMARY KEY (`nome`);
```

Tipo_esito

```
CREATE TABLE `tipo_esito` (  
  `nome` varchar(50) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
`codice` int(11) NOT NULL,  
`descrizione` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tipo_esito`  
ADD PRIMARY KEY (`codice`);
```

Tipo_locale

```
CREATE TABLE `tipo_locale` (  
  `nome` varchar(30) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Look-up Table per i tipi  
di locali';
```

```
ALTER TABLE `tipo_locale`  
ADD PRIMARY KEY (`nome`);
```

Tipo_personale

```
CREATE TABLE `tipo_personale` (  
  `nome` varchar(30) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Look-up Table per i tipi  
di personale';
```

```
ALTER TABLE `tipo_personale`  
ADD PRIMARY KEY (`nome`);
```

Tirocinio

```
CREATE TABLE `tirocinio` (  
  `id` int(11) NOT NULL,  
  `descrizione` text,  
  `idLocale` int(11) DEFAULT NULL,
```



```
`idStudente` int(11) NOT NULL,  
`idDocente` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tirocinio`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `id_tirocinio` (`id`),  
  ADD KEY `referente_tirocinio_docente` (`idDocente`),  
  ADD KEY `sede_tirocinio_locale` (`idLocale`),  
  ADD KEY `tirocinante_tirocinio_studente` (`idStudente`);
```

```
ALTER TABLE `tirocinio`  
  ADD CONSTRAINT `referente_tirocinio_docente` FOREIGN KEY (`idDocente`) REFERENCES `docente` (`idDocente`) ON UPDATE CASCADE,  
  ADD CONSTRAINT `sede_tirocinio_locale` FOREIGN KEY (`idLocale`) REFERENCES `locale` (`id`) ON UPDATE CASCADE,  
  ADD CONSTRAINT `tirocinante_tirocinio_studente` FOREIGN KEY (`idStudente`) REFERENCES `studente` (`matricola`) ON UPDATE CASCADE;
```

Turni_personale

```
CREATE TABLE `turni_personale` (  
  `id` int(11) NOT NULL,  
  `inizio` datetime NOT NULL,  
  `fine` datetime NOT NULL,  
  `plesso` varchar(50) DEFAULT NULL,  
  `piano` int(11) DEFAULT NULL,  
  `idLocale` int(11) DEFAULT NULL,  
  `idPersonale` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `turni_personale`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `id_turno_personale` (`id`),  
  ADD KEY `contratto_turno_personale` (`idPersonale`),  
  ADD KEY `contratto_sede_locale` (`idLocale`),  
  ADD KEY `contratto_sede_plesso` (`plesso`);
```

```
ALTER TABLE `turni_personale`  
  ADD CONSTRAINT `contratto_sede_locale` FOREIGN KEY (`idLocale`)
```

```
REFERENCES `locale` (`id`) ON DELETE NO ACTION ON UPDATE CASCADE,
  ADD CONSTRAINT `contratto_sede_plesso` FOREIGN KEY (`plesso`)
REFERENCES `plesso` (`nome`) ON DELETE NO ACTION ON UPDATE CASCADE,
  ADD CONSTRAINT `contratto_turno_personale` FOREIGN KEY (`idPersonale`)
REFERENCES `personale` (`idPersonale`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

Trigger

Turni_personale [BEFORE INSERT]: si assicura l'inserimento esclusivo di valori per plesso e piano o di idLocale in base al tipo di personale coinvolto nell'insert. idLocale deve essere diverso da NULL, al contrario di Plesso e Piano, per i turni del personale di tipo amministrativo, viceversa per tutti gli altri tipi di personale.

Turni_ricevimento

```
CREATE TABLE `turni_ricevimento` (
  `id` int(11) NOT NULL,
  `inizio` datetime NOT NULL,
  `fine` datetime NOT NULL,
  `idDocente` int(11) NOT NULL,
  `idLocale` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `turni_ricevimento`
  ADD PRIMARY KEY (`id`),
  ADD KEY `id_turni_ricevimento` (`id`),
  ADD KEY `ricevimento_turni_docente` (`idDocente`),
  ADD KEY `ricevimento_docente_ufficio` (`idLocale`);
```

```
ALTER TABLE `turni_ricevimento`
  ADD CONSTRAINT `ricevimento_docente_ufficio` FOREIGN KEY (`idLocale`)
REFERENCES `locale` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `ricevimento_turni_docente` FOREIGN KEY (`idDocente`)
REFERENCES `docente` (`idDocente`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

Trigger

insert_ricevimento_ufficio_docente [BEFORE INSERT]: si assicura che in ogni insert il campo *idLocale* faccia riferimento all'ufficio associato al docente identificato dal campo *idDocente*.

Back-end

Il Back-end è stato scritto in php e il suo scopo principale è quello di effettuare richieste al database e restituirne i risultati in formato json.

Ogni file php ha funzioni distinte:

calendario.php

Restituisce l'elenco delle entry nel '**calendario**' e le informazioni del relativo **locale** senza mostrare le entry riguardanti eventi passati.

config.php

Raccoglie i dati di collegamento al database.

enter_locale.php

Riceve **id_locale** tramite GET e effettua l'insert su '**log**' dell'utente attualmente loggato (utilizza i dati di sessione) e restituisce l'**esito** e la **descrizione**.

get_dati_sessione.php

Restituisce i dati di sessione dell'utente, come **nome**, **tipo_utente**, **id_utente** e **locale_presenza**

home.php

Se richiamato reindirizza alla **user_page** relativa al tipo di utente loggato.

lezioni.php

Restituisce l'elenco delle **lezioni** future o in svolgimento. Se richiesto da un docente restituisce l'elenco delle proprie lezioni, altrimenti restituisce tutte le lezioni.

list_dispositivi.php

Riceve **id_locale** tramite GET e restituisce i **dispositivi** associati al **locale**. L'utente loggato deve avere i permessi di amministratore.

list_locali.php

Restituisce tutti i **dispositivi di accesso** e tutte le informazioni del relativo **locale**.

list_tirocinio.php

Restituisce l'elenco dei **tirocini**. Se richiesto da uno studente restituisce i propri tirocini, se richiesto da un docente restituisce i tirocini in cui viene indicato come relatore.

list_utenti.php

Restituisce la lista degli utenti nel database se l'utente loggato ha i permessi di amministratore.

locali.php

Restituisce la lista dei **locali** nel database se l'utente loggato ha i permessi di amministratore.

log.php

Riceve tramite GET **tipo_id**, **inizio**, **fine**, ovvero un campo formato da tipo utente e il suo id separati da “_”, e due campi per indicare l'intervallo di tempo. Restituisce i **log** nell'intervallo di tempo per quell'utente. Se **tipo_id** = 0_0 allora restituisce i log per tutti gli utenti. Se **inizio** e **fine** sono entrambi uguali a 0 restituisce i log indipendentemente dall'intervallo di tempo. I log vengono restituiti solamente se l'utente loggato ha i permessi di amministratore.

ricevimento.php

Restituisce l'elenco dei **ricevimenti** futuri o in svolgimento. Se richiesti da un docente restituisce solamente i propri turni di ricevimento, se richiesti da uno studente restituisce i turni di tutti i professori.

statistiche_log.php

Riceve tramite GET **tipo_id**, **inizio**, **fine**, ovvero un campo formato da tipo utente e il suo id separati da “_”, e due campi per indicare l’intervallo di tempo. Restituisce il numero di accessi riusciti e quelli falliti per ogni locale a cui l’utente indicato ha provato ad accedere nell’intervallo di tempo. Se inizio e fine sono entrambi uguali a 0 restituisce i dati indipendentemente dall’intervallo di tempo. I dati vengono restituiti solamente se l’utente loggato ha i permessi di amministratore.

switch_dispositivo_locale.php

Riceve **id_dispositivo** tramite GET, effettua il toggle del campo "**attivo**" del dispositivo nella tabella '**dispositivo_locale**'. L’utente loggato deve avere i permessi di amministratore.

turni_personale.php

Restituisce l'elenco dei **turni** futuri o in svolgimento relativi all’utente loggato formattati in modo diverso a seconda se indicato il plesso oppure il locale di svolgimento. Se è indicato il locale il personale sarà di tipo amministrativo e verranno fornite non solo le informazioni riguardo al turno ma anche quelle relative al locale indicato.

user_login.php

Controlla le informazioni di accesso degli utenti e in caso di login effettuato con successo le conserva nei dati di sessione.

user_logout.php

Distrugge i dati di sessione e reindirizza alla pagina di login.

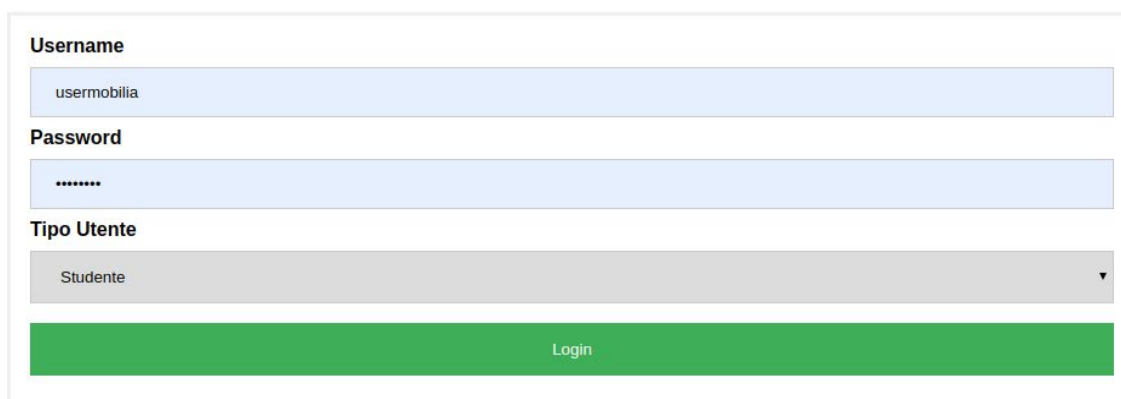
Front-end

Il Front-end è stato realizzato tramite pagine HTML supportate da tre librerie Javascript (libreria_autenticazione.js, libreria_campus.js, chart.js) ed altre due CSS (libreria_autenticazione.css, libreria_campus.css).
Ogni file html presenta una interfaccia:

index.html

la schermata iniziale presenta il form di login tramite cui i vari tipi di utente (studente, docente e personale) possono effettuare l'accesso alla piattaforma. Le credenziali di accesso consistono in uno username ed una password crittografata in md5.

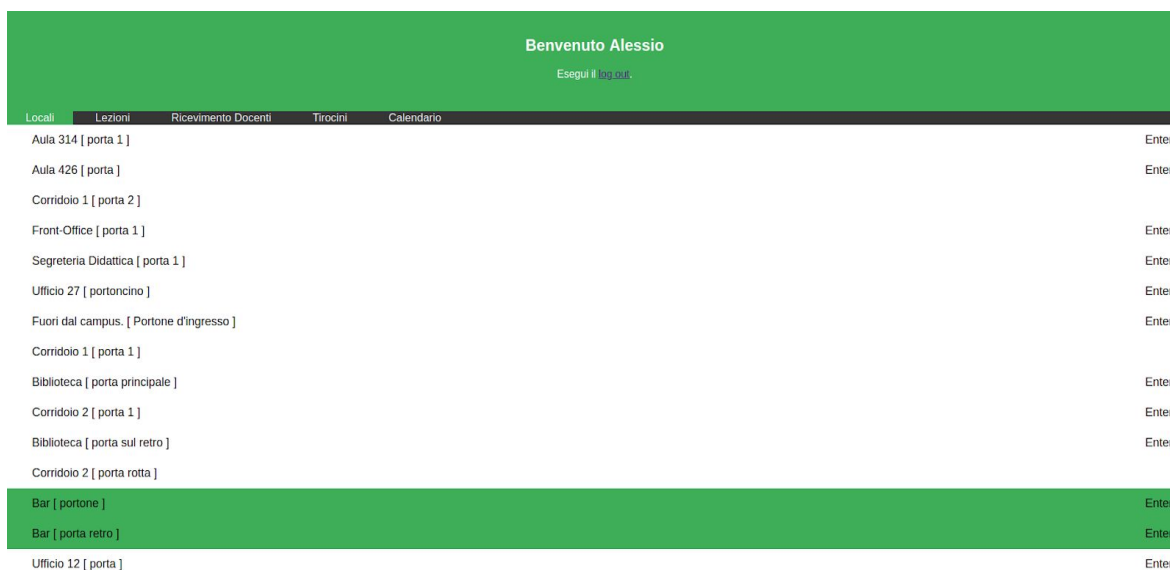
Login



The login form consists of three input fields and a button. The first field is labeled 'Username' and contains the text 'usermobilia'. The second field is labeled 'Password' and contains seven dots. The third field is labeled 'Tipo Utente' and is a dropdown menu with 'Studente' selected. Below these fields is a green button labeled 'Login'.

user_page_docente.html, user_page_personale.html, user_page_studente.html

non appena effettuato l'accesso l'utente interagire con questa interfaccia che simula lo spostamento dello stesso attraverso i locali del campus. Per ogni locale vengono mostrati

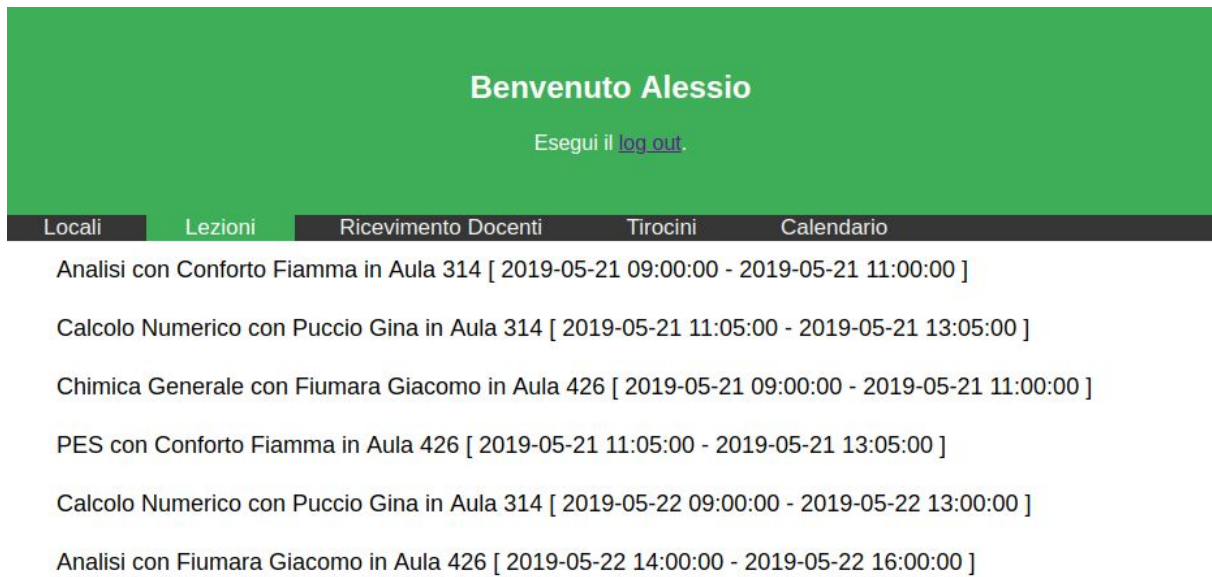


The user interface after login features a green header bar with the text 'Benvenuto Alessio' and a link 'Esegui il login'. Below the header is a navigation menu with the following items: 'Locali', 'Lezioni', 'Ricevimento Docenti', 'Tirocini', and 'Calendario'. The 'Locali' item is highlighted. Below the navigation menu is a list of campus locations, each followed by an 'Enter' key indicator on the right. The locations are: Aula 314 [porta 1], Aula 426 [porta], Corridoio 1 [porta 2], Front-Office [porta 1], Segreteria Didattica [porta 1], Ufficio 27 [portoncino], Fuori dal campus. [Portone d'ingresso], Corridoio 1 [porta 1], Biblioteca [porta principale], Corridoio 2 [porta 1], Biblioteca [porta sul retro], Corridoio 2 [porta rotta], Bar [portone], Bar [porta retro], and Ufficio 12 [porta].

tutti i dispositivi di accesso. Il locale evidenziato è quello al cui interno si trova attualmente l'utente.

lezioni.html

permette ai docenti ed agli studenti di visionare l'elenco delle lezioni fissate. Ogni lezione è corredata dal corso di laurea a cui fa riferimento, l'orario a cui si svolge e l'ubicazione dell'aula che la ospita.



Benvenuto Alessio

Esegui il [log out](#).

Locali **Lezioni** Ricevimento Docenti Tirocini Calendario

- Analisi con Conforto Fiamma in Aula 314 [2019-05-21 09:00:00 - 2019-05-21 11:00:00]
- Calcolo Numerico con Puccio Gina in Aula 314 [2019-05-21 11:05:00 - 2019-05-21 13:05:00]
- Chimica Generale con Fiumara Giacomo in Aula 426 [2019-05-21 09:00:00 - 2019-05-21 11:00:00]
- PES con Conforto Fiamma in Aula 426 [2019-05-21 11:05:00 - 2019-05-21 13:05:00]
- Calcolo Numerico con Puccio Gina in Aula 314 [2019-05-22 09:00:00 - 2019-05-22 13:00:00]
- Analisi con Fiumara Giacomo in Aula 426 [2019-05-22 14:00:00 - 2019-05-22 16:00:00]

ricevimento_docenti.html

nel caso di un docente questa pagina mostra tutti i turni di ricevimento riguardanti il docente stesso. Nel caso di uno studente vengono mostrati i turni di ricevimento relativi ad ogni docente dell'ateneo.



Benvenuto Gina

Esegui il [log out](#).

Locali **Lezioni** **Ricevimento Docenti** Tirocini Calendario

- Ufficio 27 [2019-05-21 16:00:00 - 2019-05-21 19:00:00]
- Ufficio 27 [2019-05-17 09:00:00 - 2019-05-17 13:00:00]
- Ufficio 27 [2019-05-14 15:00:00 - 2019-05-19 19:00:00]
- Ufficio 27 [2019-05-16 08:00:00 - 2019-05-16 14:00:00]

tirocini.html

questa pagina mostra tutti i tirocini associati all'utente connesso. Il docente ha accesso alle informazioni riguardanti lo studente coinvolto e viceversa.



Benvenuto Davide

Esegui il [log_out](#).

Locali Lezioni Ricevimento Docenti **Tirocini** Calendario

Analisi dei risultati. - Conforto Fiamma - Ingegneria

calendario.html

restituisce l'elenco dei locali speciali con i relativi orari di attività.



Benvenuto Davide

Esegui il [log_out](#).

Locali Lezioni Ricevimento Docenti Tirocini **Calendario**

Bar [2019-05-12 09:00:00 - 2019-05-31 21:00:00]

Front-Office [2019-05-19 08:00:00 - 2019-05-25 20:00:00]

Biblioteca [2019-05-20 07:00:00 - 2019-05-20 21:00:00]

Biblioteca [2019-05-21 08:00:00 - 2019-05-24 20:00:00]

Segreteria Didattica [2019-05-22 09:00:00 - 2019-05-22 14:00:00]

Segreteria Didattica [2019-05-22 16:00:00 - 2019-05-22 20:00:00]

turni_personale.html

mostra tutti i turni previsti dal contratto dei componenti del personale universitario. Le informazioni in questione comprendono il nome e l'ubicazione del locale associato al turno insieme a gli orari in cui quest'ultimo si svolge



The screenshot shows a web interface with a green header and a dark navigation bar. The header contains a welcome message and a link to log out. The navigation bar includes links for 'Locali', 'Turni Personale', 'Calendario', 'Log', 'Dispositivi', and 'Istogramma'. Below the navigation bar, there is a list of shifts with their respective locations and time ranges.

Benvenuto Paolo

Esegui il [log_out](#).

Locali **Turni Personale** Calendario Log Dispositivi Istogramma

- Segreteria Didattica - [2019-05-23 09:00:00 - 2019-05-23 12:00:00]
- Segreteria Didattica - [2019-05-22 12:00:00 - 2019-05-22 17:00:00]
- Front-Office - [2019-05-21 09:00:00 - 2019-05-21 13:00:00]

log.html

permette ad un utente del personale con poteri da amministratore di visionare lo storico dell'interazione tra gli utenti ed i locali del campus. La selezione può essere limitata ad un singolo utente o anche a determinati intervalli di tempo.

The screenshot shows a web interface with a green header. The header contains the text "Benvenuto Paolo" and a link "Esegui il [log out](#)". Below the header is a navigation bar with tabs: "Locali", "Turni Personale", "Calendario", "Log" (highlighted), "Dispositivi", and "Istogramma".

Below the navigation bar is a search form with a text input field containing "Alessio Mobilia". Below the input field are two date-time selection fields: "Da: 01/04/2019, 00:00" and "A: 31/05/2019, 00:00". To the right of these fields is a green "Log" button.

Below the search form is a list of log entries, each showing the user name, location, and timestamp in parentheses:

- Alessio Mobilia [studente] - Front-Office (2019-05-04 18:22:47)
- Alessio Mobilia [studente] - Fuori dal campus. (2019-05-04 18:26:45)
- Alessio Mobilia [studente] - Aula 314 (2019-05-04 18:42:24)
- Alessio Mobilia [studente] - Aula 426 (2019-05-04 18:42:25)
- Alessio Mobilia [studente] - Front-Office (2019-05-04 18:42:27)
- Alessio Mobilia [studente] - Aula 314 (2019-05-04 18:42:38)
- Alessio Mobilia [studente] - Aula 426 (2019-05-04 18:42:40)
- Alessio Mobilia [studente] - Fuori dal campus. (2019-05-07 12:00:18)
- Alessio Mobilia [studente] - Ufficio 27 (2019-05-07 12:00:25)
- Alessio Mobilia [studente] - Fuori dal campus. (2019-05-07 12:00:30)
- Alessio Mobilia [studente] - Ufficio 27 (2019-05-07 12:01:12)
- Alessio Mobilia [studente] - Segreteria Didattica (2019-05-07 12:02:20)
- Alessio Mobilia [studente] - Fuori dal campus. (2019-05-07 12:02:22)

dispositivi.html

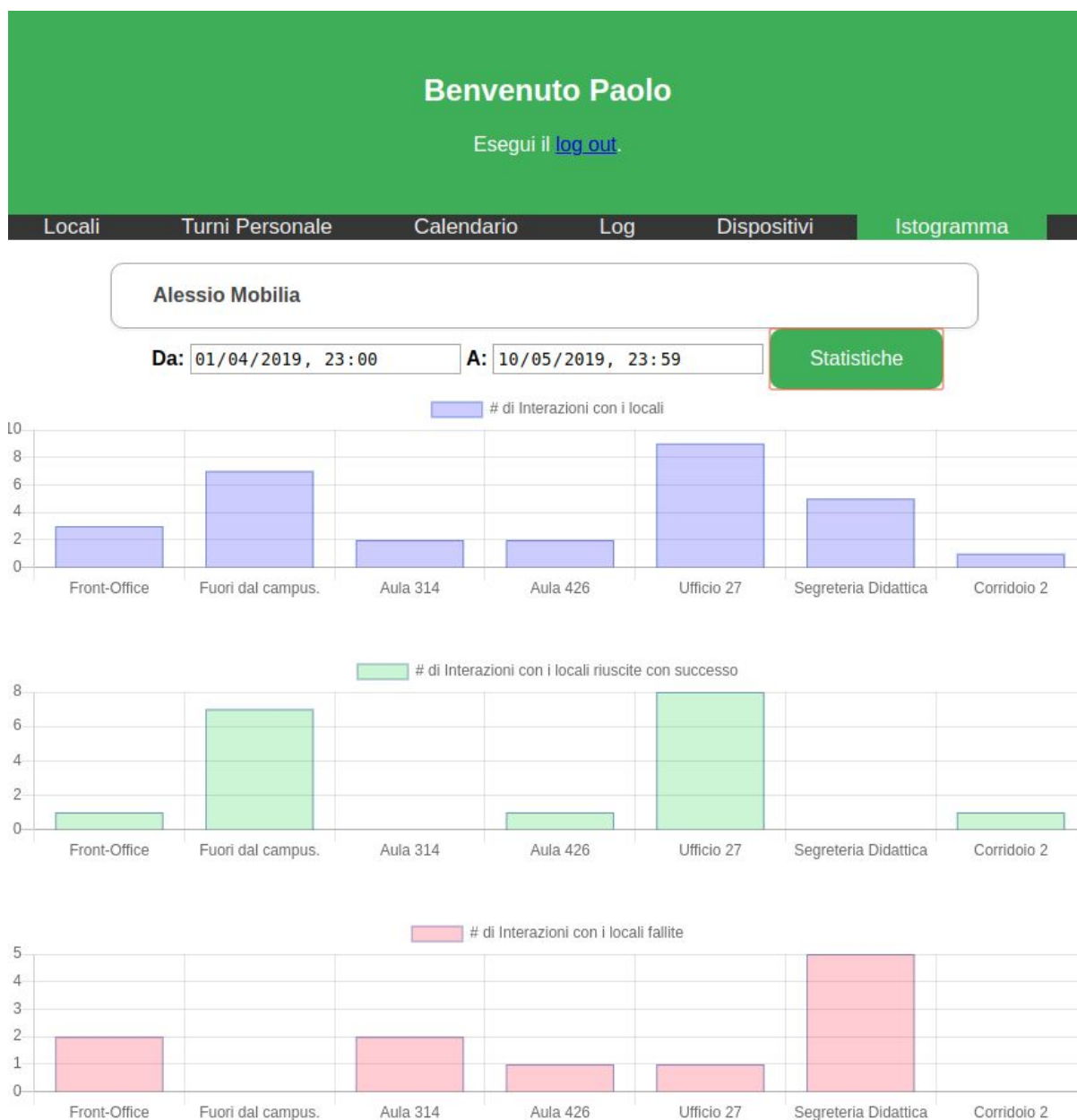
permette ad un utente del personale con poteri da amministratore di visionare l'elenco dei dispositivi presenti all'interno del campus, anche per singolo locale, ed interagire con essi tramite un pulsante apposito.

The screenshot displays a web interface for a user named Paolo. At the top, a green header contains the text "Benvenuto Paolo" and a link to "Esegui il [log out](#)". Below this is a dark navigation bar with tabs for "Locali", "Turni Personale", "Calendario", "Log", "Dispositivi" (which is highlighted), and "Istogramma". A search bar labeled "Bar" is positioned below the navigation. A green button labeled "Mostra dispositivi" is located below the search bar. The main content area lists four devices, each with a name and a function in brackets, and a corresponding green "OFF" button on the right:

- Bar: portone [Accesso] OFF
- Bar: porta retro [Accesso] OFF
- Bar: lampadario 1 [Illuminazione] OFF
- Bar: lampadario 2 [Illuminazione] OFF

istogramma.html

permette ad un utente del personale con poteri da amministratore di visionare tre istogrammi relativi all'interazione di un utente con i locali del campus. I tre istogrammi sono associati alle interazioni totali con i dispositivi d'accesso, quelle riuscite e quelle fallite. Anche qui la selezione può essere filtrata in base a particolari intervalli di tempo. Per disegnare i grafici abbiamo utilizzato la libreria open source chart.js



Sommario

SmartCampus	1
Utenza	1
Edifici	1
Dispositivi	2
Servizi	2
Accesso ai locali	2
Interazione con i dispositivi	3
Documentazione del diagramma E-R	3
Diagramma concettuale	11
Diagramma concettuale ottimizzato	12
Diagramma Logico	13
Database	14
Calendario	14
Trigger	14
Corso_di_laurea	14
Dipartimento	15
Dispositivi_utente	15
Trigger	16
Dispositivo_locale	16
Docente	17
Trigger	17
Lezione	17
Trigger	18
Locale	18
Trigger	19
Log	19
Trigger	20
Materia	20
Personale	21
Plesso	21
Presenza	22
Trigger	22
Studente	23
Tipo_dispositivi_locale	23
Tipo_esito	23
Tipo_locale	24
Tipo_personale	24

Tirocinio	24
Turni_personale	25
Trigger	26
Turni_ricevimento	26
Trigger	27
Back-end	27
calendario.php	27
config.php	27
enter_locale.php	27
get_dati_sessione.php	27
home.php	27
lezioni.php	28
list_dispositivi.php	28
list_locali.php	28
list_tirocinio.php	28
list_utenti.php	28
locali.php	28
log.php	28
ricevimento.php	28
statistiche_log.php	29
switch_dispositivo_locale.php	29
turni_personale.php	29
user_login.php	29
user_logout.php	29
Front-end	30
index.html	30
user_page_docente.html, user_page_personale.html, user_page_studente.html	30
lezioni.html	31
ricevimento_docenti.html	31
tirocini.html	32
calendario.html	32
turni_personale.html	33
log.html	34
dispositivi.html	35
istogramma.html	36
Sommario	37