***Innovative Buzzi***

|  |  |  |
| --- | --- | --- |
| Nome | Classe | Anno scolastico |
| Federico Giacumbo | 5°Q | 2019-2020 |
| Davide Hu |
| Claudiu-Cassian Ferencz |

|  |  |  |  |
| --- | --- | --- | --- |
| Versione | Data | Modifiche effettuate | Autore |
| 1.0 | 08/04/2020 | Prima versione | Federico Giacumbo |

**Descrizione dell’esigenza e piano progetto**

1. **Scopo del progetto**

Adattare la gestione della sala stampa del Buzzi con l’ausilio di nuove tecnologie.

1. **Perimetro ed ambiti di intervento**

La realizzazione di questo progetto aumenta la produttività della sala stampa riducendo al minimo i tempi di attesa.

1. **Vincoli**

Consegnare il progetto entro il 30 aprile 2020

1. **Deliverables attesi**

Struttura database realizzato, Applicazione web tramite browser.

1. **Team di progetto / gruppo di lavoro**

Federico Giacumbo = P. M gruppo di progetto

Claudiu-Cassian Ferencz = Membro del gruppo di progetto

Davide Hu = Membro del gruppo di progetto



1. **Piano delle attività**

****

****

1. **Costi di progetto**

* 2 Computer per poter lavorare contemporaneamente
* Web server per ospitare il sito web
* Database

**Analisi dei requisiti**

1. **Situazione as-is**

Nell’attuale situazione per richiedere una stampa bisogna recarsi fisicamente alla sala stampa compilando l’apposito modulo con classe, richiedente(nome cognome), numero copie necessarie, a pagamento o gratuito( per l’insegnante è gratuito) ed eventuali note. Dunque, si dovrà aspettare il proprio turno davanti alla sala stampa e attendere che vengano fatte le copie.

Diagramma di flusso interfunzionale:



1. **Situazione to-be**

Dopo la realizzazione di questo sito web sarà possibile mandare il file da stampare a distanza, riducendo tempo inutile in coda. Per usufruire di tale servizio telematico, sarà necessario fare il login (gli account verranno già creati per tutti) altrimenti sarà comunque possibile stampare recandosi fisicamente in sala stampa. Per ciascun acquisto effettuato, l’operatore registrerà il tutto nel sito cosi sarà possibile tenere traccia di tutti gli acquisti.

Digramma dei casi d’uso:



1. **Analisi degli impatti organizzativi**

Per il corretto funzionamento del sito web, sarà necessaria la presenza di due operatori: uno che si occuperà delle richieste di stampa e di registrare gli acquisti mentre l'altro svolgerà l'azione di stampa. È consigliato partecipare a dei corsi di durata di circa 2-3 ore, organizzati dal team di sviluppo per comprendere a pieno il funzionamento del sito che sarà completo al suo rilascio.

1. **Requisiti di dominio**

Tutti gli studenti e i professori potranno accedere al sito web in quanto gli account verranno creati per tutti. Potrà essere richiesto di cambiare i dati di accesso direttamente alla segreteria.

**Analisi e progettazione della base di dati**

**Progettazione concettuale**

**Dizionario delle entità**

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Sinonimo** | **Descrizione** | **Collegamento** |
| Persona | // | Persona che fa la prenotazione per la stampa online | Stampa, Prenotazione, Prodotto, Acquisto |
| Professore | // | Sottoentità di “Persona”, professore che fa la prenotazione online |  |
| Studente | // | Sottoentità di “Persona”, studente che fa la prenotazione online | Stampa |
| Prenotazione | // | Prenotazione di una stampa effettuata mandando la richiesta da sito | Persona, File |
| File | // | Contenuto che deve essere stampato presente all’interno della prenotazione | Prenotazione |
| Operatore | // | Sotto entità di Persona, Membro del personale che si occupa di stampare | Stampa |
| Stampa | // | Fotocopia del materiale (online/fisico) da stampare | Persona, Operatore, Formato, Studente |
| Formato | // | Formato della stampa che ne determina il costo | Stampa |
| Prodotto | // | Materiale che è possibile comprare oltre alle fotocopie | Acquista |
| Acquisto | // | Una persona può effettuare un acquisto | Persona,Prodotto |

**Dizionario dei dati**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Attributo** | **Tipo** | **Descrizione** |
| Persona | codiceFiscale | char(16) | Identificativo della persona (Chiave primaria) |
| email | varchar(64) | email con la quale la persona verrà identificato |
| password | varchar(256) | password della persona criptata. |
| nome | varchar(32) | nome della persona |
| cognome | varchar(32) | cognome della persona |
| dataNascita | DATE | data di nascita della persona |
| indirizzo | varchar(32) | Attributo composto formato da via, civico, Cap |
| username | varchar(64) | Username dell’utente per accedere al sito |
| Professore | // | // | // |
| Studente | classe | TINYINT | classe dello studente |
| sezione | varchar(3) | sezione della classe dello studente |
| Operatore | // | // | // |
| Prenotazione | idPrenotazione | int | Identificativo della prenotazione(chiave primaria) |
| quantità | int | numero di copie richieste da stampare |
| dataPrenotazione | DATE | data in cui si è effettuata la prenotazione |
| oraPrenotazione | time | ora in cui si è effettuata la prenotazione |
| stampata | ENUM | indica se la prenotazione è stata stampata o meno(“si” significa che è stata stampata, “no” significa che non è stata stampata) |
| File | idFile | int | Identificativo del file (chiave primaria) |
| nomeFile | varchar(128) | nome del file |
| dimensione | int | dimensione del file |
| Stampa | idStampa | int | Identificativo della stampa(chiave primaria) |
| dataStampa | DATE | data in cui la stampa ha avuto luogo |
| oraStampa | time | ora in cui la stampa ha avuto luogo |
| dataRitiro | DATE | Data in cui è stato ritirata la stampa ( se non è ancora stata ritirata sarà a null) |
| Formato | costoStampa | decimal(3,2) | costo della stampa |
| tipo | varchar(16) | tipo del formato (chiave primaria) |
| Prodotto | idProdotto | int | Identificativo del prodotto (chiave primaria) |
| nomeProdotto | varchar(64) | nome del prodotto venduto in sala stampa |
| costoProdotto | decimal(4,2) | prezzo del prodotto venduto in sala stampa |
| Acquista | idAcquisto | int | Identificativo univoco,auto\_increment (chiave primaria) |
| dataAcquisto | DATE | Data in cui viene effettuato l’acquisto |
| orarioAcquisto | TIME | Ora in cui è stato effettuato l’acquisto |
| Quantità | int | Quantità di un prodotto acquistata |

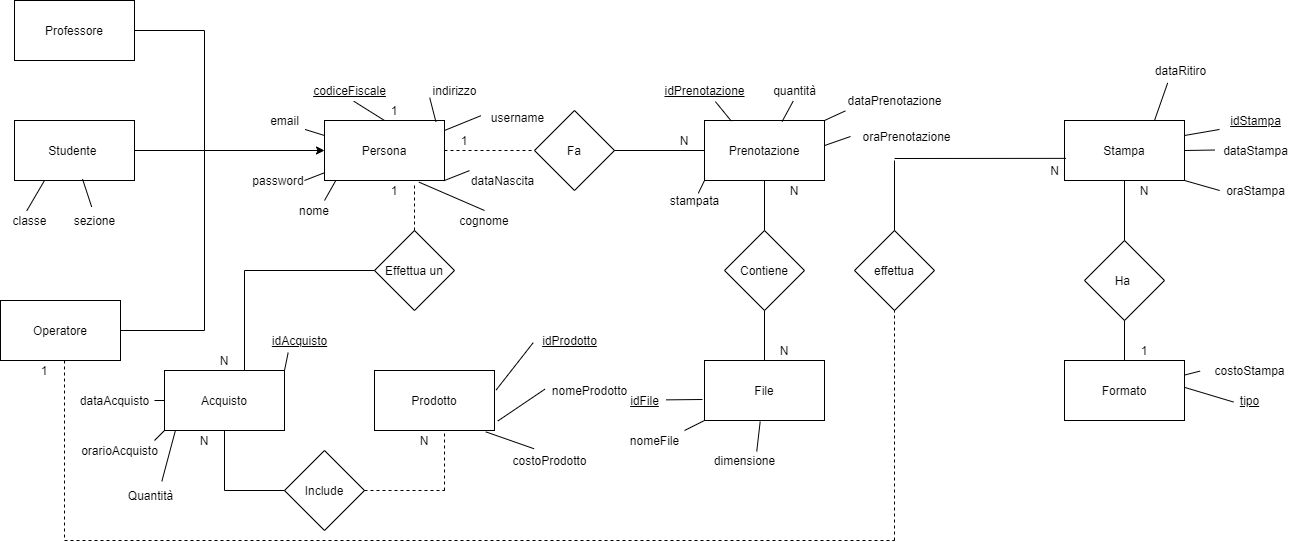
**Dizionario delle associazioni**

|  |  |  |  |
| --- | --- | --- | --- |
| **Associazione** | **Entità coinvolte** | **Descrizione** | **Attributi** |
| Fa | Persona, prenotazione | La persona fa la prenotazione per stampare | // |
| Contiene | Prenotazione, file | La prenotazione contiene il file da stampare | // |
| Effettua | Stampa, operatore | L’operatore effettua la stampa del materiale ricevuto | // |
| Ha | Stampa, formato | La stampa ha un formato | // |
| Effettua un | Acquisto-Persona | Una persona effettua un acquisto | // |
| Include | Acquisto-Prodotto | Un acquisto include uno o più prodotti | // |

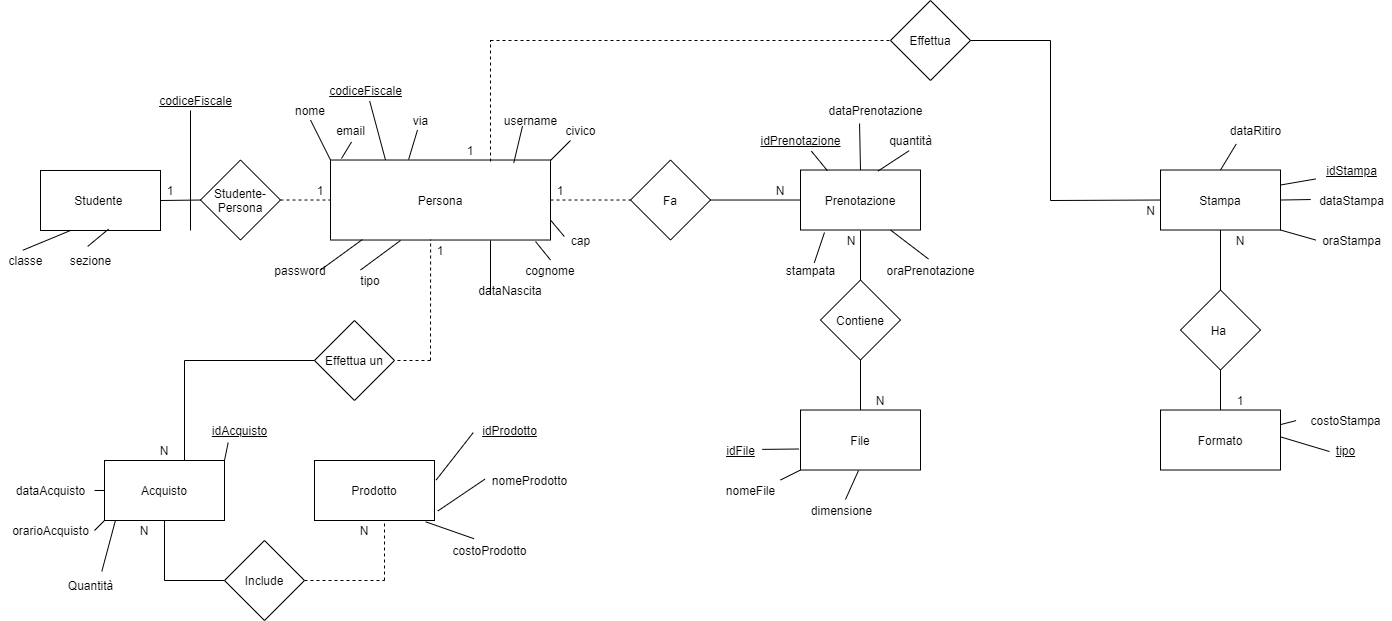
**Ipotesi aggiuntive**

* Gli utenti (professori e studenti) sono stati già registrati in quanto si usano le credenziali del vecchio registro. Se non si usassero vecchi dati, perché eliminati per il nuovo registro, verranno creati dei dati nuovi a ciascun studente e professore così da consentire l’utilizzo del servizio ai soli membri interni della scuola. In questo modo è possibile salvare e tenere traccia dei prodotti acquistati.
* Lo studente avrà come attributi classe e sezione, non essendo necessario creare delle tabelle per la gestione delle classi. Quindi per gestire copie gratuite garantite per ciascuna classe ogni anno verrà fatto a livello di software.

**Modello ER**

****

**Ristrutturazione modello ER**

****

Nella ristrutturazione abbiamo deciso di applicare una soluzione ibrida. Abbiamo eliminato le entità figlie “Professore” e “Operatore” e abbiamo inserito un boolean nell’entità “Persona”(tipo). Abbiamo lasciato l’entità figlia “Studente” in quanto possiede due attributi in più e partecipa ad una relazione in più rispetto al padre, quindi abbiamo creato una relazione chiamata “Studente-Persona” e abbiamo messo come chiave primaria di “Studente” la chiave esterna di ”Persona”.

**Progettazione Logica**

**Schema logico:**

Persona(codicefiscale,nome,cognome,username,email,password,tipo,dataNascita,

civico,cap)

Studente(codiceFiscale\*,classe,sezione)

Prodotto(idProdotto,nomeProdotto,costoProdotto)

Stampa(idStampa,dataStampa,oraStampa ,dataRitiro,oraRitiro,tipoFormato\*, codiceFiscaleOperatore\*,)

Formato(tipo,costoStampa)

Prenotazione(idPrenotazione,dataPrenotazione,quantità,oraPrenotazione,stampata,codiceFiscale\*)

File(idFile,nomeFile,dimensione)

Acquisto(idAcquisto,dataAcquisto,orarioAcquisto,Quantità,codiceFiscale\*)

Include(idAcquisto\*,idProdotto\*)

Contiene(idPrenotazione\*,idFile\*)

**Progettazione fisica**

**Query**

Create database innovativeBuzzi;

Use innovativeBuzzi;

Create table Prodotto(

idProdotto int auto\_increment primary key,

nomeProdotto varchar(64),

costoProdotto decimal(4,2)

);

create table Persona(

codiceFiscale char(16) primary key,

nome varchar(32),

cognome varchar(32),

password varchar(256),

username varchar(64),

email varchar(64),

tipo varchar(64),

dataNascita DATE,

civico int,

cap int

);

Create table Studente(

codiceFiscale char(16),

classe char(3),

sezione TINYINT,

FOREIGN KEY(codiceFiscale) REFERENCES Persona(codiceFiscale)

ON UPDATE CASCADE,

Primary key(codiceFiscale)

);

Create table Formato(

Tipo varchar(16) primary key,

costoStampa decimal(3,2)

);

Create table Stampa(

idStampa int auto\_increment primary key,

dataStampa DATE,

oraStampa TIME,

codiceFiscalePersona char(16),

dataRitiro DATE,

oraRitiro TIME,

tipoFormato varchar(16),

FOREIGN KEY(codiceFiscalePersona) REFERENCES Persona(codiceFiscale),

FOREIGN KEY(tipoFormato) REFERENCES formato(tipo)

ON UPDATE CASCADE

);

Create table acquisto(

idAcquisto int auto\_increment primary key,

codiceFiscale char(16),

dataAcquisto DATE,

orarioAcquisto TIME,

Quantità int,

FOREIGN KEY(codiceFiscale) REFERENCES Persona(codiceFiscale)

ON UPDATE CASCADE

);

create table include(

idAcquisto int,

idProdotto int,

FOREIGN KEY(idAcquisto) REFERENCES Acquisto(idAcquisto),

FOREIGN KEY(idProdotto) REFERENCES Prodotto(idProdotto)

ON UPDATE CASCADE,

primary key(idAcquisto, idProdotto)

);

Create table Prenotazione(

idPrenotazione int auto\_increment primary key,

dataPrenotazione DATE,

oraPrenotazione TIME,

quantità int,

stampata ENUM("si","no") default "no",

codiceFiscaleCliente char(16),

FOREIGN KEY(codiceFiscaleCliente) REFERENCES Persona(codiceFiscale)

ON UPDATE CASCADE

);

Create table File(

idFile int auto\_increment primary key,

nomeFile varchar(128),

dimensione int

);

Create table Contiene(

idPrenotazione int ,

idFile int,

FOREIGN KEY(idPrenotazione) REFERENCES Prenotazione(idPrenotazione),

FOREIGN KEY(idFile) REFERENCES File(idFile)

ON UPDATE CASCADE,

PRIMARY KEY (idPrenotazione, idFile)

)

**Progettazione dell’applicazione**

**Allegati**