



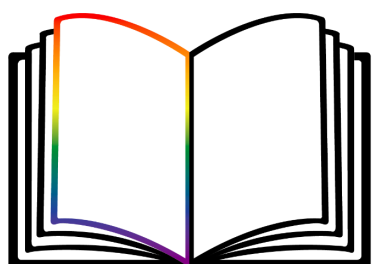
**Istituto Istruzione Superiore EINAUDI-SCARPA**  
**TECNOLOGICO - ECONOMICO - PROFESSIONALE**

Via J. Sansovino, 6 - 31044 Montebelluna (TV) • Tel. 0423-23587

[www.iiseinaudiscarpa.edu.it](http://www.iiseinaudiscarpa.edu.it)

[info@iiseinaudiscarpa.edu.it](mailto:info@iiseinaudiscarpa.edu.it) - [TVIS02400C@pec.istruzione.it](mailto:TVIS02400C@pec.istruzione.it) - [TVIS02400C@istruzione.it](mailto:TVIS02400C@istruzione.it)

Cod. Fisc.: 92036980263 Cod. Univ. Uff.: UFGZL6



# ACHILLES

*Gestione di un servizio web based per i dati e le attività di una biblioteca  
(con eventuale collegamento con altre biblioteche del territorio)*

GUSATTO DEREK

## Indice

Introduzione .....	3
Analisi dei requisiti .....	4
Pianificazione del progetto .....	5
Progettazione del database .....	8
Progettazione dell'infrastruttura di rete .....	10
Realizzazione .....	14
Creazione del database e delle relazioni .....	14
Caricamento dati nel database .....	16
Creazione delle funzionalità .....	17
<i>achilles.php</i> .....	17
<i>prestiti.php</i> .....	21
<i>nuovoprestito.php</i> .....	23
<i>insertprestito.php</i> .....	24
<i>creapdf.php</i> .....	25
<i>mail.php</i> .....	27
Conclusioni .....	29

## **Introduzione:**

Lo scopo del progetto è la creazione di un'applicazione web in grado di gestire i dati e le attività delle biblioteche biblioteca appartenenti a un sistema bibliotecario dislocato sul territorio e il loro collegamento.

L'esperienza che più mi ha ispirato nell'ideazione di questo progetto è l'attività di PCTO svolta nel corso del terzo anno presso il Comune di Possagno grazie alla quale ho potuto imparare, lavorando nella Biblioteca comunale, ad utilizzare il programma gestionale usato dal Polo Bibliotecario di Castelfranco Veneto per la gestione delle singole biblioteche e del sistema bibliotecario nel suo complesso. Questa esperienza mi sarà utile, nell'implementazione della piattaforma e nella progettazione dell'infrastruttura di rete, per capire come gestire efficacemente le interfacce e le funzionalità anche dal punto di vista dei bibliotecari, tenendo in considerazione le loro necessità e comodità.

Per questo il punto centrale del progetto sarà quindi la creazione di un'interfaccia web semplice e intuitiva che permetta di effettuare prestiti, inserire titoli a catalogo e inserire nuovi utenti nel sistema in pochi click. La prima funzionalità da sviluppare è quindi una homepage che permetta la visualizzazione dei libri presenti nella Biblioteca presso cui sono registrati gli utenti o presso la quale lavorano i bibliotecari, e che permetta ai primi di visualizzare la propria situazione prestiti e modificare il proprio profilo e ai secondi di gestire il catalogo (aggiungendo, modificando o eliminando libri), gli utenti e i prestiti, potendo scegliere se visualizzare solo prestiti scaduti o rientrati.

Nel progettare l'infrastruttura verranno distinte le autorizzazioni di bibliotecari e utenti della biblioteca. Sarà presente un server per il sito della Biblioteca e una rete wi-fi per i dispositivi mobili degli utenti. Solo le postazioni dei bibliotecari, poi, potranno accedere alla rete interbibliotecaria attraverso una connessione protetta.

Particolare attenzione deve essere data anche alla sicurezza della rete interna alla Biblioteca, controllando gli accessi e le minacce, sia interne che esterne.

## **Analisi dei requisiti:**

All'interno del database sarà necessario salvare i dati relativi ai libri presenti all'interno della biblioteca (comprensivi di titolo, anno di pubblicazione, lingua, numero di pagine, prezzo, ISBN, autore, editore genere e un eventuale immagine della copertina) con la relativa collocazione e degli utenti registrati (credenziali di accesso e informazioni di contatto, informazioni anagrafiche e indirizzo). Nella base di dati sarà presente inoltre uno storico dei prestiti per salvare data di inizio e fine, se il prestito è rientrato o è stato prorogato.

Sarebbe efficace implementare l'applicazione con l'invio di alcuni file in formato pdf via mail riassuntivi del prestito appena registrato.

L'infrastruttura di rete della biblioteca deve prevedere diverse postazioni, differenziate attraverso delle VLAN, separando così le postazioni dei bibliotecari da quelle degli utenti. Sarà presente anche un Access-Point per garantire agli utenti della biblioteca di poter sfruttare l'accesso internet della biblioteca.

Nella rete saranno presenti anche un DHCP-server per l'assegnazione dinamica degli indirizzi IP ai dispositivi wireless e un web-server per permettere l'accesso alla pagina web della biblioteca, debitamente protetti. L'accesso ai server e alla rete sarà controllato con delle ACL.

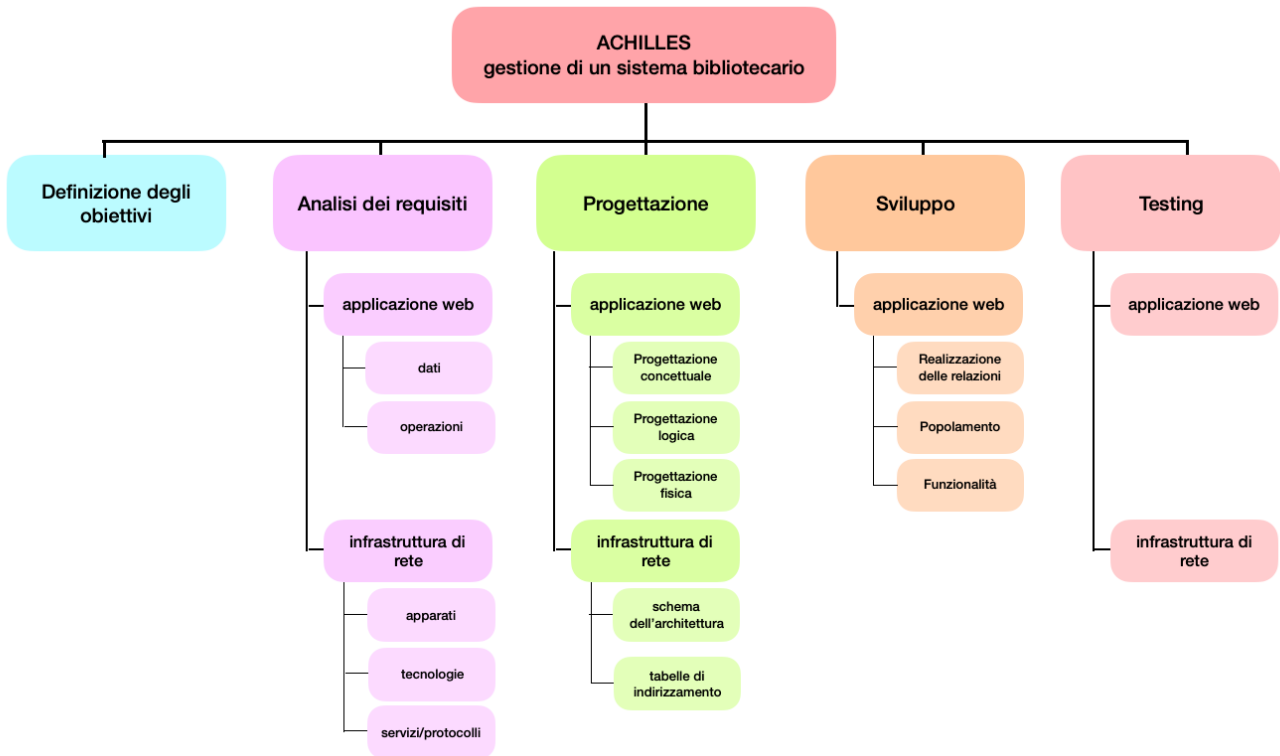
Infine, per garantire il collocamento della biblioteca all'interno della rete bibliotecaria sul territorio, sarebbe opportuno impostare una VPN per una connessione sicura con le altre biblioteche.

L'accesso ad internet infine avverrà mascherando gli indirizzi IP della rete grazie al protocollo NAT statico.

Nella realizzazione del progetto verrà data maggiore priorità ad una corretta progettazione del database così da prevenire inconvenienti futuri, avendo anche un'occhio di riguardo alla normalizzazione.

Per quanto riguarda l'infrastruttura di rete, invece, la maggiore priorità sarà data alla corretta gestione delle autorizzazioni di accesso a dispositivi e rete dall'esterno.

## Pianificazione del progetto: WBS - Work Breakdown Structure



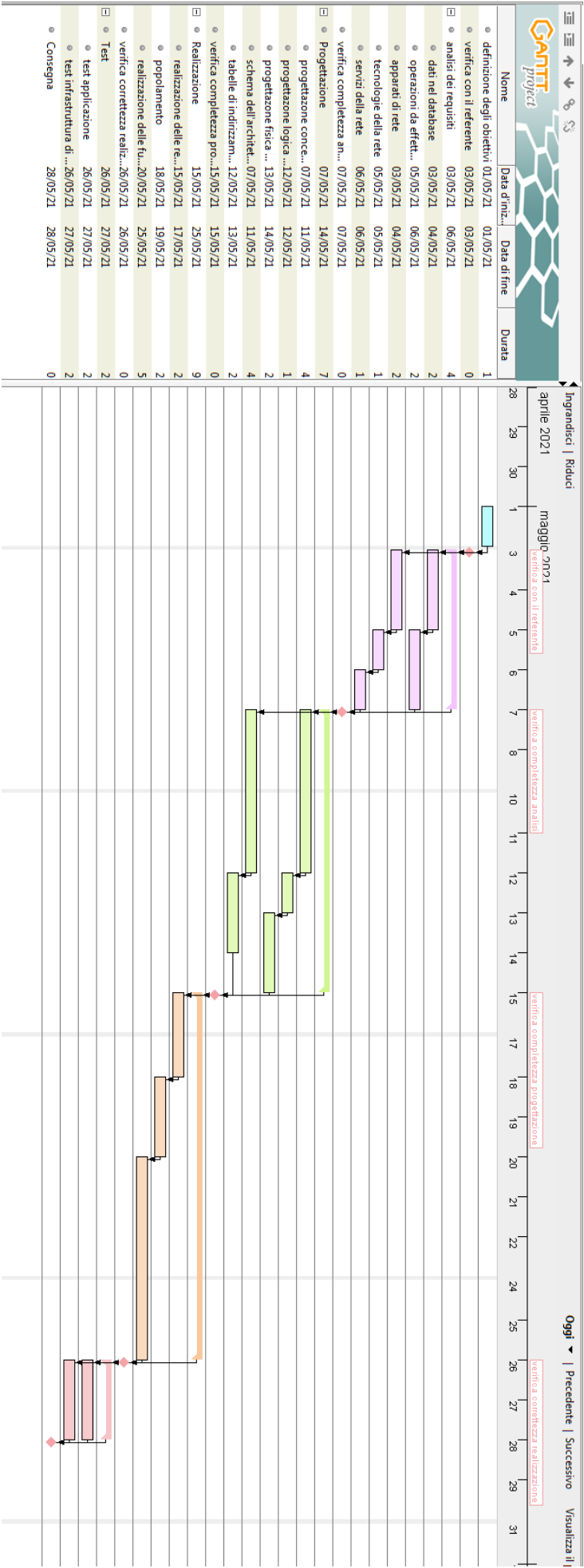
La WBS aiuta a scomporre il progetto in sub componenti e/o i relativi rilasci progressivi (deliverable) e rappresenta in modo strutturato e gerarchico tutte le attività che richiedono tempo e risorse, utilizzando il metodo “top down” in modo molto simile alla programmazione strutturata.

I work-package di questa WBS sono i classici componenti del progetto informatico: definizione degli obiettivi, analisi, progettazione, realizzazione e testing.

In questo caso è inoltre abbastanza evidente la suddivisione tra processi legati all'applicazione web e processi legati all'infrastruttura di rete.

Ovviamente la fase di documentazione procederà lungo tutta la durata del progetto.

GANTT

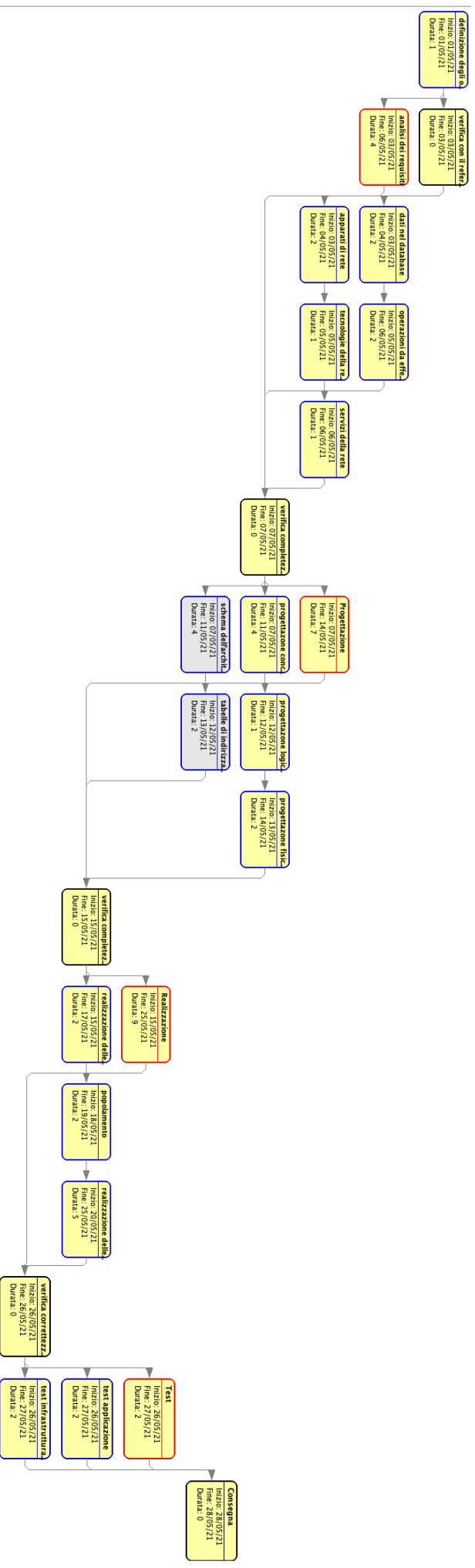


Il diagramma di GANTT mostra la pianificazione temporale del progetto in ogni sua attività evidenziando la sequenzialità di alcune attività e la possibilità di eseguire alcune operazioni in contemporanea.

Nel GANTT sono presenti dei punti di verifica, utili a controllare lo stato di avanzamento delle attività come previsto fino a quel momento.

In questo GANTT risulta molto evidente il parallelismo tra le attività legate all'applicazione e quelle utili alla realizzazione dell'infrastruttura di rete.

## PERT - Project Evaluation and Review Technique

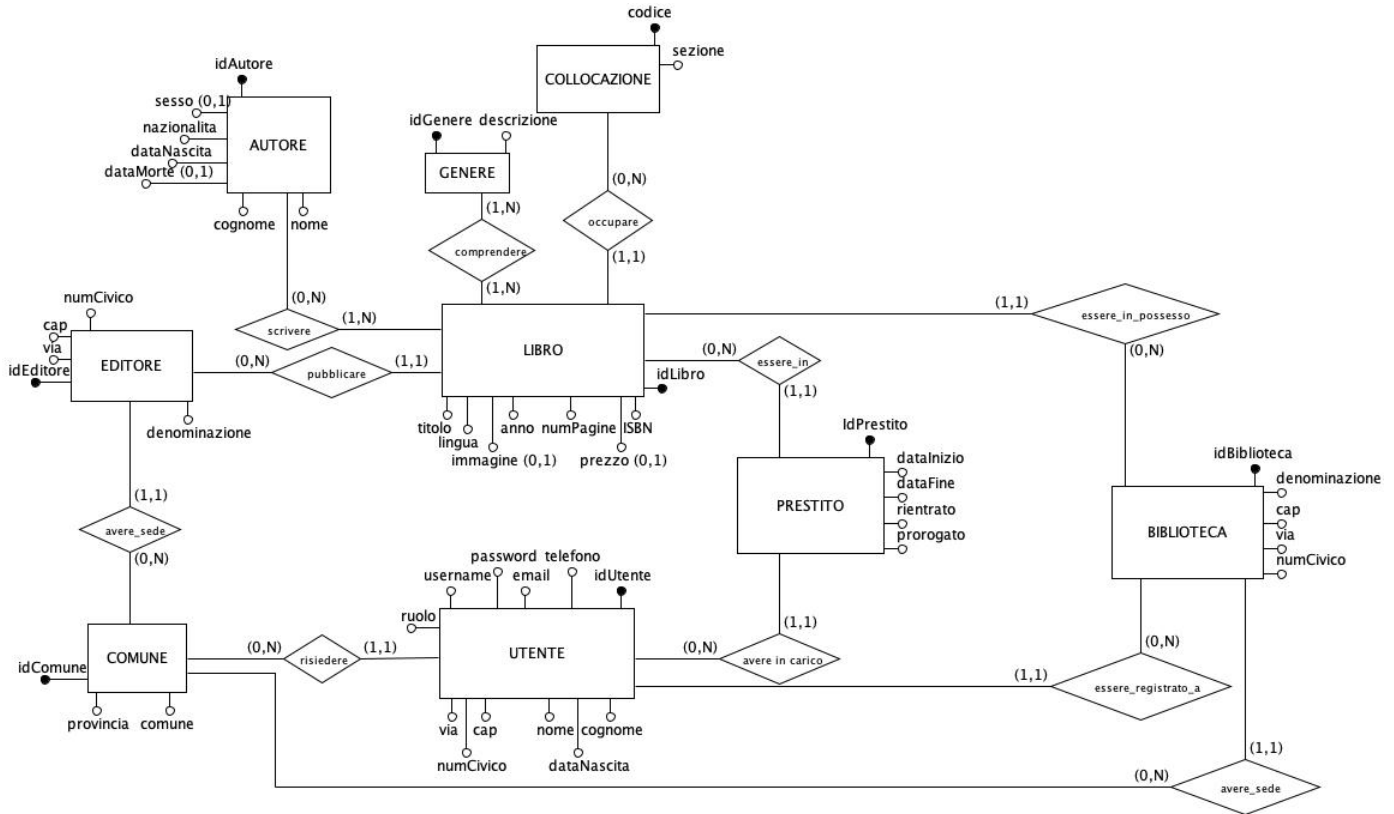


Il PERT pone in evidenza la successione delle attività con un grafo unidirezionale aciclico che descrive il progetto nella sua interezza, indicando le interdipendenze tra le attività.

Nel diagramma PERT è inoltre visibile il cammino (o percorso) critico determinato attraverso le tecniche del *CPM - Critical Path Method*.

## Progettazione del database:

### Progettazione concettuale del database:



Nell'entità *Utente*, *username* sarà il nome univoco utilizzato dall'utente per accedere e la *password* sarà salvata con codifica SHA1 e impostata di default uguale allo username.

*Ruolo* rappresenta il livelli di privilegi dell'utente, 'bibliotecario' oppure 'user'.

Nella relazione *Libro* vi sono degli attributi facoltativi: campi *prezzo* e *immagine* e quest'ultimo servirà a contenere il path dell'immagine di copertina del libro.

Anche all'interno della tabella *Autore* *dataMorte* e *sex* sono dei campi facoltativi.

*Indirizzo* all'interno di *Utente*, *Biblioteca* e *Editore*, pensato come un campo composto, viene già decomposto nei suoi attributi atomici per attenersi alla Prima Forma Normale.

Saranno generate dalla progettazione logica due relazioni: *Scrivere*, nata dalla relazione N:N *scrivere* tra *Libro* e *Autore*, e *Comprendere*, dalla relazione N:N *comprendere* tra *Libro* e *Genere*.



## Progettazione logica del database:

```
COMUNE(idComune, comune, provincia)
EDITORE(idEditore, denominazione, via, numCivico, cap)
BIBLIOTECA(idBiblioteca, denominazione, via, numCivico, cap)
AUTORE(idAutore, nome, cognome, dataNascita, dataMorte (NULL), sesso
      (NULL), nazionalita)
GENERE(idGenere, descrizione)
COLLOCAZIONE(codice, sezione)
LIBRO(idLibro, ISBN, prezzo(NULL), numPagine, anno, immagine (NULL),
      lingua, titolo, fkBiblioteca, fkCollocazione, fkEditore)
UTENTE(idUtente, username (UNIQUE), password, email, telefono,
      ruolo, via, numCivico, cap, nome, cognome, dataNascita,
      fkComune, fkBiblioteca)
PRESTITO(idPrestito, dataInizio, dataFine, rientrato, prorogato,
      fkLibro, fkUtente)
SCRIVERE(fkAutore, fkLibro)
COMPRENDERE(fkGenere, fkLibro)
```

## Progettazione fisica del database:

Sono state scelte delle chiavi primarie numeriche intere *AUTO\_INCREMENT* (fatta esclusione per *codice* in *Collocazione* e per le chiavi primarie composte nelle tabelle *Scrivere* e *Comprendere* ) per appesantire il meno possibile gli indici di chiave primaria che nel DBMS utilizzato comportano anche l'ordinamento delle tabelle.

Per quanto riguarda le transazioni invece il DBMS utilizzato utilizza tecniche di locking che portano a un livello di isolamento di tipo *repeatable read*.

## Progettazione dell'infrastruttura di rete:

Nella rete sono state create delle **VLAN** (Virtual LAN) ovvero delle LAN realizzate logicamente grazie allo standard 802.1Q che permette di creare più VLAN che utilizzano e condividono una stessa infrastruttura fisica pur restando delle reti separate dove i pacchetti broadcast sono confinati all'interno di essa. Le VLAN create sono due:

- VLAN 10 - BIBLIOTECA, a cui sono collegati gli host dei bibliotecari e la stampante di rete
- VLAN 20 - UTENTI, a cui sono collegate le postazioni degli utenti e gli eventuali dispositivi wireless che si connettono alla rete

Le due VLAN comunicano tra loro grazie alla tecnologia denominata "Router-on-a-stick" che permette di "suddividere" un'interfaccia fisica in più interfacce virtuali (subinterfacce) che possono comunicare tra loro. Ogni subinterfaccia collegata a una VLAN e come indirizzo IP il default gateway della VLAN.

Per facilitare il controllo e la sicurezza della rete sono poi state formate due **subnet**, una per la VLAN 10 (192.168.1.0/25) e una per la VLAN 20 (192.168.1.128/25).

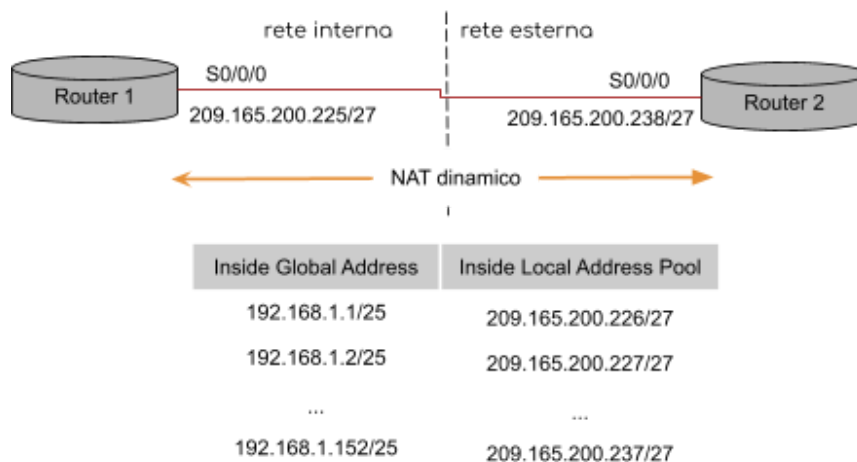
Nella rete sono stati collegati due **server**:

- SERVER WEB & DNS (172.17.100.100): gestisce dei clienti con protocollo HTTP e HTTPS della pagina *www.achilles.com* che rappresenta ipoteticamente l'applicazione web realizzata. All'interno del server è attivo anche il servizio DNS che traduce il nome simbolico del sito nel suo indirizzo IP per le richieste al server.
- SERVER DHCP (172.17.100.101): permette l'assegnazione dinamica dell'indirizzo IP ai dispositivi che si connettono, tramite l'Access-Point, alla rete. I possibili indirizzi da assegnare sono racchiusi in un pool denominato *wireless* che comprende gli indirizzi da 192.168.1.132/25 a 192.168.1.152/25 con default-gateway 192.168.1.254/25 e default DNS 172.17.100.100/16

Per la protezione dei server viene creata **ACL** - Access Control List - estesa con nome (*SERVER\_PROTECTION*) che permetta il passaggio delle richieste al server DHCP, permetta la navigazione verso il server web utilizzando i protocolli HTTP e HTTPS. Tra le entry dell'ACL vengono inoltre inserite quelle che permettono le connessioni *established* e le richieste al DNS, oltre a quella che nega le risposte ai *ping*.

Una seconda ACL (*OUTSIDE*) limita il traffico entrante dall'esterno della rete permettendo solo le richieste HTTP e HTTPS al server web.

Nell'ambito dell'interazione con la rete esterna inoltre viene implementato il **NAT** - Network Address Translation - dinamico che permette alle VLAN di uscire dalla rete interna con un indirizzo IP pubblico scelto, secondo in modalità FIFO, tra quelli predefiniti nel pool di indirizzi del NAT.



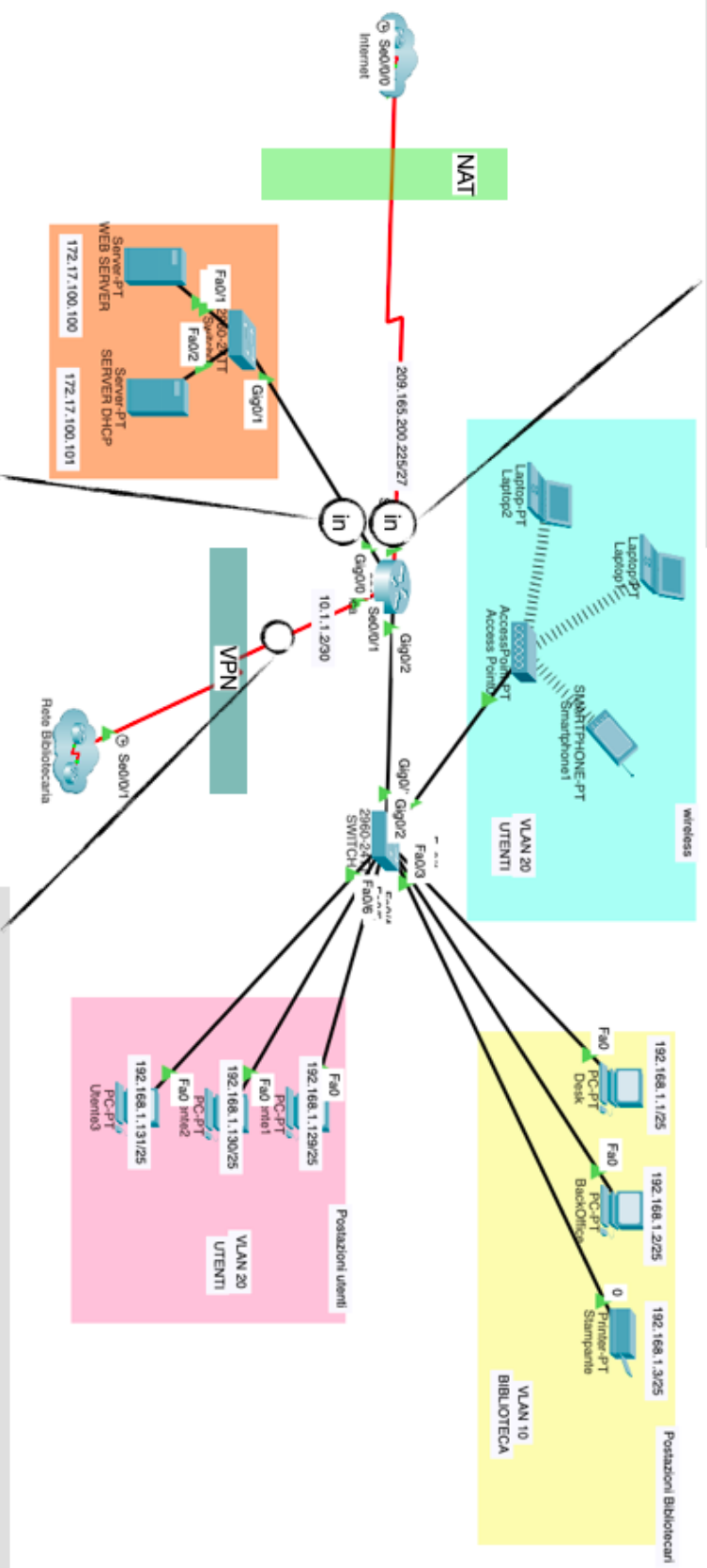
Per l'interconnessione con altre biblioteche si ipotizza la creazione di una rete più ampia alla quale la rete della biblioteca di conetterà tramite una **VPN** per creare una connessione privata end-to-end tramite un accesso strettamente controllato. Utilizzando una VPN realizzata grazie alla tecnologia *IPsec* le informazioni provenienti da una rete privata vengono trasportare in modo sicuro su una rete pubblica. L'*IPsec* risponde a quattro funzioni critiche di riservatezza (crittografia), integrità dei dati,



autenticazione e protezione anti-replay (prevenire pacchetti duplicati). Questo tipo di VPN sfrutta il protocollo *IKE* (Internet Key Exchange) e la configurazione prevede due fasi. La prima è la definizione delle policy di *IKE* (utilizzando però il comando *ISAKMP* - Internet Security Association and Key Management Protocol, rimasto dal protocollo usato precedentemente, su cui si basa l'*IKE*) definendo la modalità di crittografia (*AES*), il tipo di autenticazione (pre-shared key) e il gruppo dell'algoritmo Diffie-Hellman (*group 2*) usato per stabilire la chiave segreta condivisa tra peer. La seconda parte di configurazione è necessaria a stabilire il *transform-set*: dopo avergli assegnato un nome è necessario indicare la crittografia da utilizzare per l'*ESP* - Encapsulating Security Payload (3DES, SHA, ...). Occorre infine creare una mappa che imposti l'indirizzo del peer destinatario e che utilizzi il *transform-set* appena creato, associando ad essa una ACL che permetta il passaggio degli indirizzi che devono sfruttare il canale sicuro. Essendo una biblioteca pubblica è infine essenziale permettere agli utenti della biblioteca di poter connettere i propri dispositivi **wireless** alla rete. A tale scopo viene collegato un Access-Point per propagare la rete con SSID *Biblioteca* e autenticazione con WPA2-PSK e criptata con l'algoritmo AES. La rete wireless sarà propagata sul Canale 6 a 2.4GHz e avrà copertura nel raggio di 100m.

## Infrastruttura di rete e configurazione:

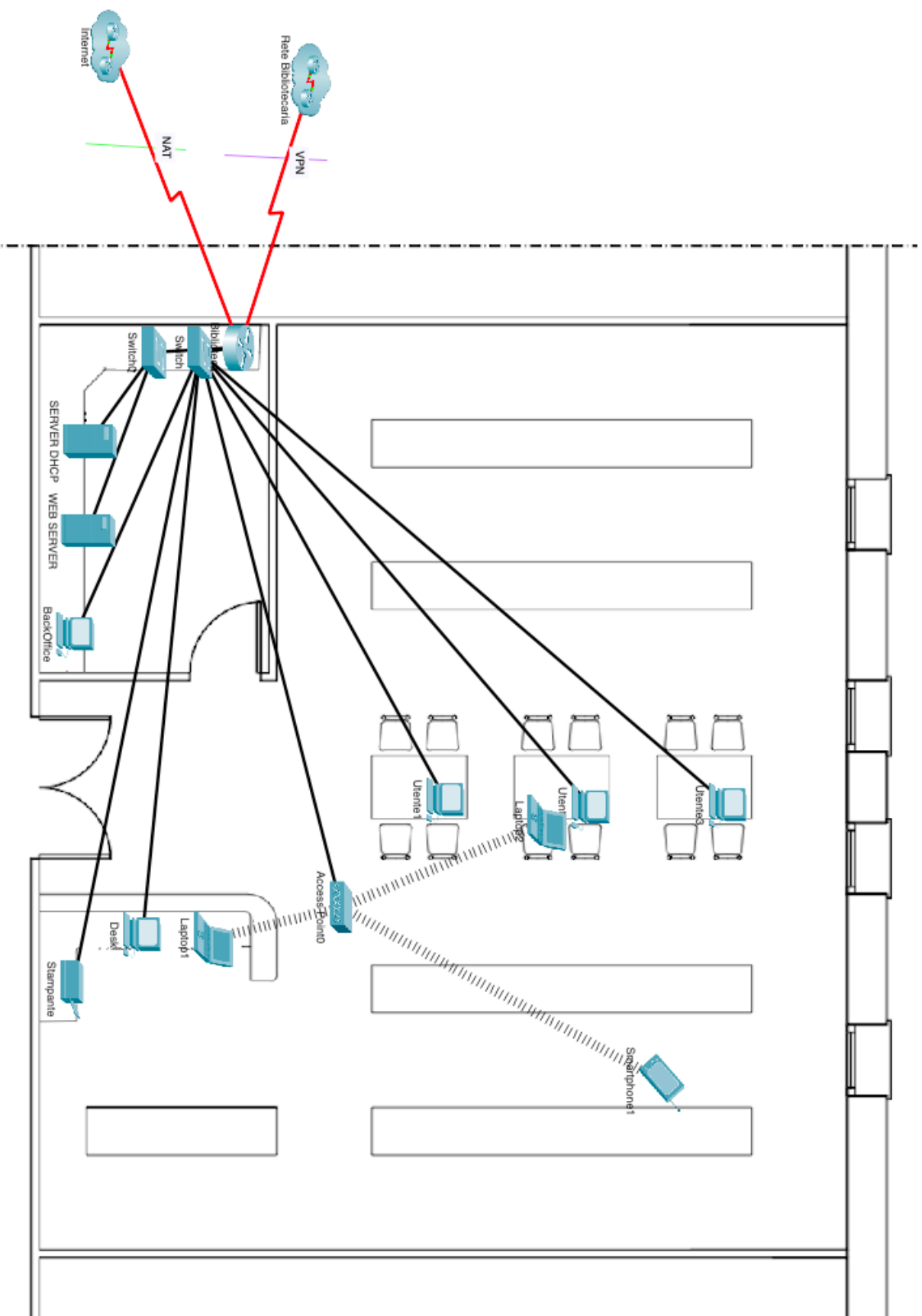
```
ip access-list extended OUTSIDE
 permit tcp any host 172.17.100.100 eq www
 permit tcp any host 172.17.100.100 eq 443
```



```
ip access-list extended SERVER_PROTECTION
remark perimetrete richieste DHCP
permit udp any any eq bootpc
permit udp any any eq bootps
remark perimetrete navigazione verso server web
permit tcp any 172.17.100.100 0.0.255.255 eq www
permit tcp any 172.17.100.100 0.0.255.255 eq 443
remark perimetrete connessioni established
permit tcp any any established
remark perimetrete DNS
permit tcp any eq domain any
remark bloccare risposte a ping
deny icmp any any echo
```

```
crypto isakmp policy 10
  encr aes
  authentication pre-share
  group 2
crypto isakmp key biblioteca address 10.1.1.1
crypto ipsec transform-set VPN esp-3des esp-sha-hmac
crypto map VPN-MAP 10 ipsec-isakmp
description VPN connessione rete interbibliotecaria
set peer 10.1.1.1
set transform-set VPN
match address 101
```

Possibile disposizione della rete all'interno dei locali della biblioteca:



## Realizzazione:

### Creazione del database:

```
CREATE DATABASE `biblioteca`  
DEFAULT CHARACTER SET utf8;
```

### Creazione delle relazioni:

```
CREATE TABLE `libro` (  
  `idLibro` int(11) NOT NULL PRIMARY KEY,  
  `titolo` varchar(100) NOT NULL,  
  `lingua` varchar(100) NOT NULL,  
  `immagine` varchar(150) DEFAULT NULL,  
  `anno` int(11) NOT NULL DEFAULT (YEAR(CURRENT_TIMESTAMP())),  
  `numPagine` int(11) NOT NULL,  
  `prezzo` decimal(6,2) NOT NULL CHECK(prezzo > 0),  
  `ISBN` varchar(13) NOT NULL,  
  `fkEditore` int(11) NOT NULL,  
  `fkCollocazione` int(11) NOT NULL,  
  `fkBiblioteca` int(11) NOT NULL,  
  FOREIGN KEY (fkBiblioteca) REFERENCES biblioteca(idBiblioteca),  
  FOREIGN KEY (Collocazione) REFERENCES collocazione(idCollocazione),  
  FOREIGN KEY (fkEditore) REFERENCES editore(idEditore)  
)  
  
CREATE TABLE `utente` (  
  `idUtente` int(11) NOT NULL PRIMARY KEY,  
  `username` varchar(40) NOT NULL UNIQUE,  
  `password` varchar(40) NOT NULL,  
  `email` varchar(100) NOT NULL,  
  `nome` varchar(100) NOT NULL,  
  `cognome` varchar(100) NOT NULL,  
  `dataNascita` date NOT NULL CHECK(dataNascita < CURRENT_TIMESTAMP()),  
  `telefono` varchar(10) NOT NULL,  
  `via` varchar(50) NOT NULL,  
  `numCivico` varchar(4) NOT NULL,  
  `cap` varchar(5) NOT NULL,  
  `ruolo` varchar(20) NOT NULL DEFAULT 'user',  
  `fkComune` int(11) NOT NULL,  
  `fkBiblioteca` int(11) NOT NULL,  
  FOREIGN KEY (fkBiblioteca) REFERENCES biblioteca(idBiblioteca),  
  FOREIGN KEY (fkComune) REFERENCES comune(idComune)  
)
```

```

CREATE TABLE `biblioteca` (
  `idBiblioteca` int(11) NOT NULL PRIMARY KEY,
  `denominazione` varchar(50) NOT NULL,
  `via` varchar(40) NOT NULL,
  `numCivico` varchar(4) NOT NULL,
  `cap` varchar(5) NOT NULL,
  `fkComune` int(11) NOT NULL,
  FOREIGN KEY (fkComune) REFERENCES comune(idComune)
)

CREATE TABLE `prestito` (
  `idPrestito` int(11) NOT NULL PRIMARY KEY,
  `dataInizio` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP(),
  `dataFine` datetime NOT NULL,
  `rientrato` tinyint(1) NOT NULL DEFAULT 0,
  `prorogato` tinyint(1) NOT NULL DEFAULT 0,
  `fkLibro` int(11) NOT NULL,
  `fkUtente` int(11) NOT NULL,
  FOREIGN KEY (fkLibro) REFERENCES libro(idLibro),
  FOREIGN KEY (fkUtente) REFERENCES utente(idUtente),
  CHECK (dataFine>dataInizio)
)

```

Nella creazione vera e propria delle tabelle all'interno di *phpMyAdmin* dovranno essere tolti i vincoli di dominio (*CHECK*) e alcuni valori di default a causa della parziale aderenza del DBMS allo standard SQL2.

## Caricamento dati nel database:

```
INSERT INTO `biblioteca` (`idBiblioteca`, `denominazione`, `via`, `numCivico`,  
`cap`, `fkComune`) VALUES  
(1, 'Biblioteca di Montebelluna', 'Largo Dieci Martiri', '1', '31044', 5),  
(2, 'Biblioteca di Possagno', 'Via Molinetto', '10', '31054', 6);
```

```
INSERT INTO `libro` (`idLibro`, `titolo`, `lingua`, `immagine`, `anno`,  
`numPagine`, `prezzo`, `ISBN`, `fkEditore`, `fkCollocazione`, `fkBiblioteca`)  
VALUES  
(1, 'Io non ho paura', 'italiano', NULL, 2011, 230, '11.00', '9788806207694', 3,  
111, 2),  
(2, 'Aristotele e Dante scoprono i segreti dell\'universo ', 'inglese', NULL,  
2012, 358, '10.00', '9781442408937', 7, 222, 2),  
(3, 'La canzone di Achille', 'italiano', NULL, 2013, 379, '12.00',  
'9788807893469', 5, 222, 2),  
(4, 'Colpa delle Stelle', 'italiano', NULL, 2015, 347, '13.00', '9788817081566',  
6, 333, 2),  
(5, 'La guerra degli Elfi', 'inglese', NULL, 2013, 315, '10.00', '9788804530268',  
3, 222, 2),  
(6, 'The hate you give', 'inglese', NULL, 2017, 410, '14.00', '9788809836402', 7,  
222, 2),  
(7, 'Il fu Mattia Pascal', 'italiano', NULL, 2014, 302, '9.00', '9788806219116',  
2, 111, 1),  
(8, 'La coscienza di Zeno', 'italiano', NULL, 2014, 364, '13.00', '9788806222444',  
2, 111, 1);
```

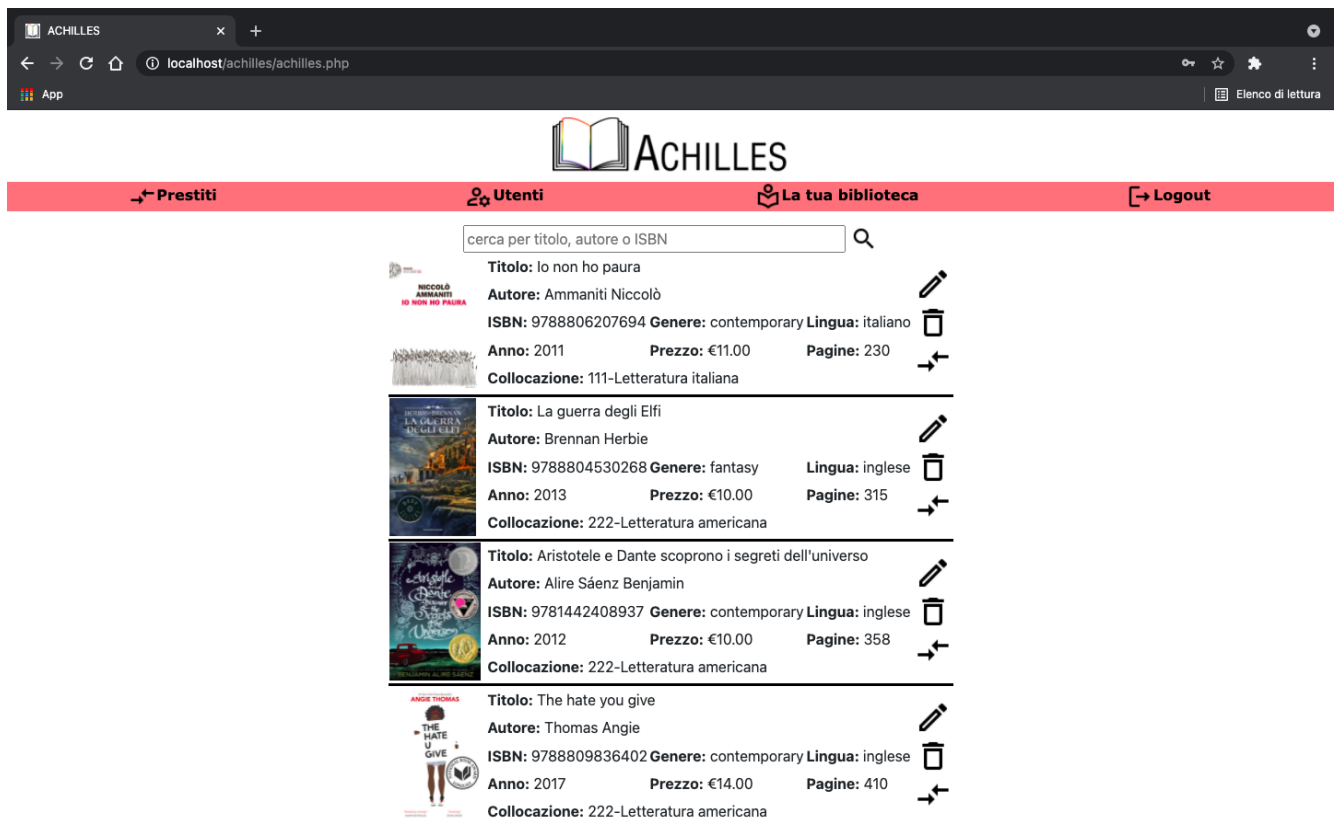
```
INSERT INTO `utente` (`idUtente`, `username`, `password`, `email`, `nome`,  
`cognome`, `dataNascita`, `telefono`, `via`, `numCivico`, `cap`, `ruolo`,  
`fkComune`, `fkBiblioteca`) VALUES  
(1, 'gusattoderek', 'e981335aba70fec46acb8b5df3f0b4e91d4ea589',  
'gusattoderek@gmail.com', 'Derek', 'Gusatto', '2002-06-06', '3314996861', 'via  
Cunial', '63', '31054', 'bibliotecario', 6, 2),  
(4, 'pandolfoenrico', 'a83c59fdc1868bf9c49ac1f62b1aa388797e8998',  
'enrico.pandolfo02@gmail.com', 'Enrico', 'Pandolfo', '2002-12-26', '3200767227',  
'via Chiesa Castelli', '7', '31010', 'user', 10, 2),  
(5, 'bandieramattia', '6cac480c2985025f80e437975093ec05de7f845a',  
'mattia.bandiera10@gmail.com', 'Mattia', 'Bandiera', '2002-05-10', '3405190023',  
'via della Salute', '8', '31011', 'user', 9, 1),  
(6, 'zanattaloreenzo', 'b365c51956790d0b1c3f5fde614a676630e1e55b',  
'zanataaaloreenzo@gmail.com', 'Lorenzo', 'Zanatta', '2001-06-10', '3454952060', 'via  
dei martiri', '22', '31044', 'user', 5, 1),  
(7, 'montagnermattia', '3e44b941f0004290a9683ebb1d6ce3eba3c647d7',  
'mattiamontagner2002@gmail.com', 'Mattia', 'Montagner', '2002-09-28',  
'3474307246', 'via Martinella', '26', '31044', 'bibliotecario', 5, 1);
```



## Creazione delle funzionalità:

**achilles.php**: la prima funzionalità realizzata è l'homepage dell'applicazione, visibile a utenti autenticati e non quindi l'intestazione viene creata di volta in volta con la tecnologia *ajax* richiamando la pagina *caricamentointestazione.php* che controlla, attraverso le variabili di sessione, se l'utente che visita la pagina è un utente, un bibliotecario oppure non ha effettuato il login.

Tutti gli utenti che accedono pagina possono effettuare una ricerca all'interno del catalogo, mentre solo i bibliotecari visualizzano la possibilità di modificare, eliminare o dare in prestito i libri.



```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>ACHILLES</title>
<!-- FOGLI DI STILE -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/css/bootstrap.min.css"
integrity="sha384-B0vP5xmATw1+K9KRQjQERJvTumQW0nPEzvF6L/Z6nronJ3oUOFUFpCjEUQouq2+1"
crossorigin="anonymous">
```

```

<link rel="stylesheet" href="stylesheet.css">
<!-- cambio icona scheda -->
<link rel="icon" href="images/icons/icona.png" />

<script> //Funzione javascript che richiama la pagina php che crea la giusta intestazione
della pagina con tecnologia ajax
    function intestazione() {
        var request;
        request = new XMLHttpRequest;
        request.onreadystatechange = risposta;
        request.open("post", "caricamentointestazione.php");
        request.send();
        function risposta() {
            if (request.readyState == 4 && request.status == 200) {
                document.getElementById("intestazione").innerHTML = \
                    request.responseText;
            }
        }
    }
</script>
</head>
<body onload="intestazione()">
<center>
<div class="header">
<a href="achilles.php" ></a>
<div id="intestazione"></div>
</div>
<div class="container"> <!--FORM PER LA RICERCA NEL CATALOGO-->
    <form name="cerca" method="get" action="<?php echo $_SERVER['PHP_SELF']; ?>"
style="margin-top: 15px;">
        <input size="50" type="text" name="cerca" placeholder="cerca per titolo, autore o
ISBN" style="vertical-align: middle" />
        <input type="image" src="images/icons/search.png" alt="Submit" width="30" height="30"
style="border-width: 0; vertical-align: middle" >
    </form>
</div>
<?php
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'biblioteca');
//Connessione al database
$con = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
if ($con->connect_errno) {
    printf("Connect failed: %s\n", $con->connect_error);
    exit();
}
$con->set_charset("utf8");
//verifica se sono presenti canoni di ricerca
if(isset($_GET['cerca'])){

```

```

        $cerca = $_GET['cerca'];
    }else{
        $cerca = '';
    }

    //Seleziona i libri che corrispondo ai criteri di ricerca (se presenti) e che sono
    presnti nella biblioteca presso cui è registrato l'utente (se questo haneffettuato il
    login)
    $sql = "SELECT libro.*, autore.nome, autore.cognome, collocazione.*, genere.descrizione";
    $sql .= " FROM libro";
    $sql .= " INNER JOIN scrivere ON libro.idLibro=scrivere.fkLibro";
    $sql .= " INNER JOIN autore ON autore.idAutore=scrivere.fkAutore";
    $sql .= " INNER JOIN collocazione ON libro.fkCollocazione=collocazione.codice";
    $sql .= " INNER JOIN comprendere ON libro.idLibro=comprendere.fkLibro";
    $sql .= " INNER JOIN genere ON genere.idGenere=comprendere.fkGenere";
    $sql .= " WHERE idLibro IN (SELECT idLibro FROM libro";
    $sql .= " WHERE titolo LIKE '%" . $cerca . "%'";
    $sql .= " OR autore.nome LIKE '%" . $cerca . "%'";
    $sql .= " OR autore.cognome LIKE '%" . $cerca . "%'";
    $sql .= " OR ISBN LIKE '%" . $cerca . "%' )";
    if(isset($_SESSION['idBiblioteca'])){
        $sql .= " AND idLibro IN (SELECT idLibro FROM libro INNER JOIN biblioteca ON
        idBiblioteca=fkBiblioteca WHERE idBiblioteca = '" . $_SESSION['idBiblioteca'] . "')";
    }
    $result = $con->query($sql);
    //se la query ha avuto successo, stampa i risultati
    if ($result) {
        if ($result->num_rows > 0) {
            echo "<table>";
            while($row=$result->fetch_array()){
                echo "<tr >";
                echo "<td rowspan='5' ><img src='" . $row['immagine'] . "' height='150'>
                </td>";
                echo "<td colspan='3'><b>Titolo:</b> " . $row['titolo'] . "</td>";
                //se l'utente è un bibliotecario, mostra la possibilità di modifica,
                eliminare o dare in prestito il libro
                if(isset($_SESSION['ruolo']) AND $_SESSION['ruolo']=='bibliotecario'){
                    echo "<td rowspan='5'>";
                    echo "<table>";
                    echo "<tr><td><a href='modifica.php?idLibro=" . $row['idLibro'] . "'>
                    <img src='images/icons/edit.png' height='40'></a></td></tr>";
                    //Modifica del libro
                    echo "<tr><td><a href='elimina.php?idLibro=" . $row['idLibro'] . "'>
                    <img src='images/icons/delete.png' height='40'></a></td></tr>";
                    //elimina il libro
                    echo "<tr><td><ahref='nuovoprestito.php?idLibro=" .
                    $row['idLibro'] . "'> <img src='images/icons/prestito.png'
                    height='40'></a></td></tr>"; //Prestito del libro
                    echo "</table>";
                    echo "</td>";
                }
            }
            echo "</tr>";


```

```

echo "<tr>";
echo "<td colspan='3' ><b>Autore:</b> ".$row['cognome']." ".$row['nome']
    . "</td>";
echo "</tr>";
echo "<tr>";
echo "<td ><b>ISBN:</b> ".$row['ISBN']."</td>";
echo "<td ><b>Genere:</b> ".$row['descrizione']."</td>";
echo "<td ><b>Lingua:</b> ".$row['lingua']."</td>";
echo "</tr>";
echo "<tr>";
echo "<td ><b>Anno:</b> ".$row['anno']."</td>";
echo "<td ><b>Prezzo:</b> €".$row['prezzo']."</td>";
echo "<td ><b>Pagine:</b> ".$row['numPagine']."</td>";
echo "</tr>";
echo "<tr>";
echo "<td colspan='3' ><b>Collocazione:</b> ".$row['codice']."-".
$row['sezione']."</td>";
echo "</tr>";
echo "<tr style='border-bottom: 3px solid black;'><td colspan='5'>
</td></tr>";
    }
    echo "</table>";
} else {
}
} else {
    echo "errore: ".$con->error;
}
?>
</center>
</body>
</html>

```

**prestiti.php**: una pagina accessibile dall'homepage che riporta un report completo dei prestiti in corso. È disponibile inoltre una funzione di ricerca per titolo, autore o ISBN. Una funzionalità molto utile di questa pagina consente al bibliotecario di cercare tra i prestiti scaduti oppure nello storico di prestiti già rientrati. Una variante di questa pagina (**prestiti\_utente.php**) permette agli utenti la visualizzazione dei loro prestiti in corso. L'unica differenza è nella query inviata al database la quale richiede come condizione dei prestiti da selezionare che siano a carico dell'utente, memorizzato grazie a una variabile di sessione.



### Query e visualizzazione dei prestiti:

//Creazione della SELECT SQL

```
$sql = "SELECT prestito.*, utente.nome, utente.cognome, libro.titolo, libro.ISBN,
collocazione.*, autore.nome AS anome, autore.cognome AS acognome";
$sql .= " FROM libro";
$sql .= " INNER JOIN scrivere ON libro.idLibro=scrivere.fkLibro";
$sql .= " INNER JOIN autore ON autore.idAutore=scrivere.fkAutore";
$sql .= " INNER JOIN collocazione ON libro.fkCollocazione=collocazione.codice";
$sql .= " INNER JOIN prestito ON libro.idLibro=prestito.fkLibro";
$sql .= " INNER JOIN utente ON utente.idUtente=prestito.fkUtente";
$sql .= " WHERE idLibro IN (SELECT idLibro FROM libro"; //Sub-query per la ricerca in
titolo, autore o ISBN
$sql .= " WHERE titolo LIKE '%" . $cerca . "%'";
$sql .= " OR autore.nome LIKE '%" . $cerca . "%'";
$sql .= " OR autore.cognome LIKE '%" . $cerca . "%'";
$sql .= " OR ISBN LIKE '%" . $cerca . "%' )";
if(isset($_SESSION['idBiblioteca'])) { //sub-query per la ricerca all'interno della
biblioteca presso qui l'utente è registrato
    $sql .= " AND idLibro IN (SELECT idLibro FROM libro INNER JOIN biblioteca ON
idBiblioteca=fkBiblioteca WHERE idBiblioteca = '" . $_SESSION['idBiblioteca'] . "')";
}
if($scaduti==true) { //Se si vogliono visualizzare solo prestiti scaduti
    $sql .= " AND prestito.dataFine < current_timestamp()";
} else if($rientrati==true) { //Se si vogliono visualizzare solo prestiti rientrati
    $sql .= " AND prestito.rientrato = 1";
}
```

```

}else{ //altrimenti, di default, visualizza i prestiti in corso
    $sql .= " AND prestito.rientrato = 0";
}
$result = $con->query($sql);
if ($result) {
    if ($result->num_rows > 0) {
        echo "<table>";
        while($row=$result->fetch_array()){
            echo "<tr >";
            echo "<td style='outline: thin solid; background-color: lightcoral;'><b>Prestito:</b> ".$row['idPrestito']. "</td>";
            echo "</tr>";
            echo "<tr >";
            echo "<td><b>Utente:</b> ".$row['cognome']. " ".$row['nome']. "</td>";
            echo "</tr>";
            echo "<tr>";
            echo "<td><b>Inizio:</b> ".$row['dataInizio']. "</td>";
            echo "<td><b>Fine:</b> ".$row['dataFine']. "</td>";
            //bottone per la prooga del prestito
            echo "<td><a href='proroga.php?idPrestito=".$row['idPrestito']."'><img src='images/icons/proroga.png' height='40'></a></td>";
            echo "</tr>";
            echo "<tr>";
            echo "<td><b>Libro:</b> ".$row['titolo']. "</td>";
            echo "<td><b>Autore:</b> ".$row['acognome']. " ".$row['anome']. "</td>";
            //bottone per il rientro del prestito
            echo "<td><a href='rientro.php?idPrestito=".$row['idPrestito']."'><img src='images/icons/rientro.png' height='40'></a></td>";
            echo "</tr>";
        }
        echo "</table>";
    }
} else {
    echo "errore: ".$con->error;
}

```

**nuovoprestito.php**: richiede l'inserimento di utente, libro e giorni di prestito per poi passare i dati a *insertprestito.php*. Se si arriva alla pagina da *achilles.php* allora l'id del libro sarà già salvato e verrà passato alla pagina successiva tramite un campo nascosto nel form.

---



← PrestitiUtentiLa tua bibliotecaLogout →

Utente

Libro

Giorni di prestito

Inserisci

### Creazione del form in *nuovoprestito.php*

```
<div>
    <form id="prestito" method="post" action="insertprestito.php" style="margin-top:
10px; padding-top: 5px; padding-bottom: 5px;">
    <label for="utente">Utente</label>
    <input type="text" id="utente" name="utente" list="utenti" ></br>
    <datalist id="utenti">
    <?php //creo una datalist con gli utenti della biblioteca
        $sql = "SELECT nome, cognome, idUtente FROM utente WHERE fkBiblioteca=" .
            $_SESSION['idBiblioteca'] . " ";
        $result = $con->query($sql);
        if($result){
            if($result->num_rows > 0){
                while($row=$result->fetch_array()){
                    echo "<option value=" . $row['idUtente'] . ">" . $row['cognome'] . "
". $row['nome'] . "</option>";
                }
            }
        }
    </datalist>
    <?php if(!isset($_GET['idLibro'])){ ?>
    <label for="libro">Libro</label>
    <input type="text" id="libro" name="libro" list="libri" ></br>
    <datalist id="libri">
    <?php //creo una datalist con i libri disponibili
        $sql = "SELECT titolo, idLibro FROM libro WHERE
fkBiblioteca=" . $_SESSION['idBiblioteca'] . " AND idLibro NOT IN (SELECT fkLibro FROM
prestito WHERE rientrato='0')";
        echo $sql;
        $result = $con->query($sql);
        if($result){
            if($result->num_rows > 0){
                while($row=$result->fetch_array()){
                    echo "<option value=" . $row['idLibro'] . ">" . $row['titolo'] . "</option>";

```

```

        } }}
    }else{ //Se ho già selezionato il libro per il prestito in achilles.php
    ?>
    <input type="hidden" id="libro" name="libro" list="libri" value=<?php echo $idLibro ?>>
    <?php } ?>
    </datalist>
    <label for="giorni">Giorni di prestito</label>
    <input type="number" name="giorni" step="5" min="15" max="40" value="30"></br>
    <button type="submit" class="btn btn-dark">Inserisci</button>
    </form>
</div>

```

***insertprestito.php***: si tratta di una pagina *non visibile* all'utente. Questa pagina recupera, infatti, i dati inseriti da un utente bibliotecario in *nuovoprestito.php* e li utilizza per salvare sulla base di dati il prestito appena effettuato. Dopo aver recuperato i dati inseriti dall'utente la pagina fa partire una transazione per evitare stati di inconsistenza nel database dovuti ad accessi concorrentiali o ad eventuali errori. All'interno della transazione viene controllato che il libro non sia già in prestito e che l'utente non abbia già troppi prestiti a suo carico (il numero massimo previsto è 5), successivamente viene inserito il nuovo prestito nella database e chiusa la transazione. Infine la pagina effettua un reindirizzamento a *creapdf.php*, passando come parametro il numero del prestito.

#### Gestione dell'inserimento del prestito nel database con una transazione:

```

$errore = false;
$con->autocommit(false); //non vi è un commit dopo ogni istruzione SQL
$con->begin_transaction(); //BOT
$response;
$sql = "SELECT idLibro FROM libro WHERE idLibro = '". $idLibro . "'";
$sql .= " AND idLibro NOT IN (SELECT fkLibro FROM prestito WHERE rientrato='0')";
$result = $con->query($sql);
if ($result) {
    if($result->num_rows > 0){
    }else{
        $errore=true;
        $response="Libro già in prestito";
    }
}
}
else{
    $errore=true;
    $response="Fallimento inserimento del prestito";
}
}
$sql = "SELECT COUNT(idPrestito) AS tot FROM prestito WHERE fkUtente = '". $idUtente . "'
AND rientrato = '0' GROUP BY fkUtente";
$result = $con->query($sql);
if ($result) {
    if($result->num_rows > 0){
        while($row=$result->fetch_array()){
            $tot = $row['tot']*1;
            if($tot>=5){

```



```

        $errore=true;
        $response="L'utente ha troppi prestiti in corso";
    }
}
} else{
    $errore=true;
}
$sql = "INSERT INTO `prestito` (`dataInizio`, `dataFine`, `fkLibro`, `fkUtente`)";
$sql .= " VALUES (current_timestamp(), ADDDATE(current_timestamp(), INTERVAL ".$giorni."
DAY), ".$idLibro.", ".$idUtente.")";
$result = $con->query($sql);
if ($result) {
    if($con->affected_rows > 0){
        $last_id = $con->insert_id;
    } else{
        $errore=true;
        $response="Fallimento inserimento del prestito";
    }
} else{
    $errore=true;
} //EOT
if($errore==true){ //se qualcosa è andato storto -> rollback
    $con->rollback();
    echo "Errore: ".$response;
} else{ //se è andato tutto a buon fine -> commit
    $con->commit();
    header('Location: creapdf.php?idPrestito='.$last_id);
}

```

**creapdf.php:** crea un documento in formato PDF riassuntivo dei dati di un prestito appena inserito, inserendo un'intestazione con il logo del sistema e un piè di pagina. L'ultima azione è il reindirizzamento a *mail.php*.



#### **Prestito: 31**

Biblioteca: Biblioteca di Possagno

Utente: Gusatto Derek

Inizio: 2021-05-23 11:15:09 Fine: 2022-08-16 11:15:09

ISBN: 9788806207694, ID Libro: 1

Libro: Io non ho paura

Autore: Ammaniti Niccolò

## Creazione e salvataggio del file pdf:

```
$pdf = new PDF(); // classe PDF è una FPDF modificata aggiungendo header e footer
$pdf->AddPage();
$pdf->SetFont('Times','',16);
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'biblioteca');
$con = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
if ($con->connect_errno) {
    printf("Connect failed: %s\n", $con->connect_error);
    exit();
}
$con->set_charset("utf8");
//richiede al database tutti i dati relativi al prestito, al libro e all'utente a cui
è in carico
$sql = "SELECT prestito.*, utente.idUtente, utente.nome, utente.cognome,
libro.titolo, libro.ISBN, collocazione.*, autore.nome AS anome, autore.cognome AS
acognome, biblioteca.denominazione";
$sql .= " FROM libro";
$sql .= " INNER JOIN scrivere ON libro.idLibro=scrivere.fkLibro";
$sql .= " INNER JOIN autore ON autore.idAutore=scrivere.fkAutore";
$sql .= " INNER JOIN collocazione ON libro.fkCollocazione=collocazione.codice";
$sql .= " INNER JOIN prestito ON libro.idLibro=prestito.fkLibro";
$sql .= " INNER JOIN utente ON utente.idUtente=prestito.fkUtente";
$sql .= " INNER JOIN biblioteca ON utente.fkBiblioteca=biblioteca.idBiblioteca";
$sql .= " WHERE idPrestito = '". $idPrestito. "'";
$result = $con->query($sql);
if ($result) {
    //stampa i risultati
    if ($result->num_rows > 0) {
        while($row=$result->fetch_array()){
            $idUtente = $row['idUtente'];
            $pdf->SetFont('Times','B',16);
            $pdf->Cell(40,10,utf8_decode('Nuovo Prestito: '). $idPrestito), '', 2);
            $pdf->SetFont('Times','',16);
            $pdf->Cell(40,10,utf8_decode('Biblioteca: '). $row['denominazione']), '', 2);
            $pdf->Cell(40,10,utf8_decode('Utente: '). $row['cognome']. " ". $row['nome']),
            '', 2);
            $pdf->Cell(50,10,utf8_decode('Inizio: '). $row['dataInizio']));
            $pdf->Cell(20,10,'');
            $pdf->Cell(50,10,utf8_decode('Fine: '). $row['dataFine']), '', 2);
            $pdf->SetX(10);
            $pdf->Cell(60,10,utf8_decode('Libro: '). $row['titolo']. " ". $row['ISBN']),
            '', 2);
            $pdf->Cell(60,10,utf8_decode('Autore: '). $row['acognome']. " ".
            $row['anome']), '', 2);
        }
    }
}
//salva il file nella cartella del server
```

```

$path=$_SERVER['DOCUMENT_ROOT'].'ACHILLES/pdfprestiti/prestito'.$idPrestito.'.pdf';
$name = 'prestito'.$idPrestito;
$pdf->Output('F', $path);
//richiama la pagina che invia il file all'utente via mail
header("Location: mail.php?utente=".$idUtente."&allegato=".$name);

```

**mail.php:** una pagina non visibile all'utente che invia un'email all'utente recuperando i dati del prestito e l'indirizzo email dell'utente dal database e allegando il documento appena creato da *creapdf.php*.

#### Creazione e invio della mail:

```

require './PHPMailer/src/SMTP.php';
require './PHPMailer/src/PHPMailer.php';
require './PHPMailer/src/Exception.php';

//Crea una nuova istanza
$mail = new PHPMailer();

//Usa protocollo SMTP
$mail->isSMTP();

$mail->SMTPDebug = SMTP::DEBUG_OFF;

//utilizza GMAIL
$mail->Host = 'smtp.gmail.com';

//Set del numero di porta
$mail->Port = 587;

//Set della crittografia
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;

// e dell'autenticazione
$mail->SMTPAuth = true;

//casella di posta da cui invio la mail
$mail->Username = 'maildaluca@gmail.com';

//Password
$mail->Password = 'lucamail.00';

//Indirizzo da cui deve arrivare dal mail e nome
$mail->setFrom('noreply@achilles.com', 'Achilles');

//Indirizzo di risposta alla mail e nome
$mail->addReplyTo('noreply@achilles.com', 'Achilles');

//prelevo dal database l'email dell'utente
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'biblioteca');

$con = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
if ($con->connect_errno) {
    printf("Connect failed: %s\n", $con->connect_error);
    exit();
}

$con->set_charset("utf8");

```

```

$m = '';
$sql = "SELECT email FROM utente WHERE idUtente='".$utente."'";
echo "sql: ".$sql;
$result = $con->query($sql);
if ($result) {
    if ($result->num_rows > 0) {
        $row=$result->fetch_array();
        $m=$row['email'];
    }
}
//Set email destinatario
$mail->addAddress($m, 'Utente'.$utente);
//Set oggetto della mail
$mail->Subject = 'Nuovo prestito ';
//Set corpo del messaggio
$mail->msgHTML("Nuovo prestito all'utente");
//Allegato pdf
$path = $_SERVER['DOCUMENT_ROOT'].'/ACHILLES/pdfprestiti/'.$allegato.'.pdf';
$mail->addAttachment($path, $name = 'prestito', $encoding = 'base64', $type =
'application/pdf');
//Invio
if (!$mail->send()) {
    echo 'Mailer Error: ' . $mail->ErrorInfo;
} else {
    header('Location: achilles.php');}

```

## Conclusioni:

Giunto alla fine del progetto sono decisamente soddisfatto di come sono riuscito a mettere in pratica quanto appreso in questi tre anni che studio informatica e in particolare quanto studiato nel corso dell'ultimo anno.

L'esperienza di seguire un progetto dalla sua ideazione e progettazione fino alla sua realizzazione e testing oltre ad essermi senz'altro utile per il futuro, mi ha permesso di mettermi alla prova e spingermi oltre agli esercizi più classici aiutandomi a capire come è meglio gestire un progetto sia in termini di tempo che di sviluppo graduale.

Nonostante all'inizio del progetto, nella fase di analisi, il tempo previsto per la fase di progettazione concettuale del database non fosse troppo, al momento di realizzarla ha richiesto più tempo del previsto a causa dell'inaspettata complessità dei dati da gestire. Mi sarebbe piaciuto riuscire ad implementare anche la gestione di libri e utenti nell'applicazione web, ma a causa della complessità di gestione dei prestiti e dei tempi stretti ho dovuto rinunciarci, rimandando a un possibile sviluppo futuro.

D'altro canto sono felice di aver portato a termine una procedura completa per i prestiti, fino alla creazione di un documento riassuntivo e all'invio di una mail.

Per quanto io non abbia riscontrato particolari difficoltà con i linguaggi utilizzati, ho dovuto scontrarmi con la progettazione dell'infrastruttura di rete più complessa in cui mi sia mai cimentato, imbattendomi spesso in complicate configurazioni di apparati di rete che richiedevano vari tentativi e molto troubleshooting.

Un ultimo punto che ha risentito dei tempi stretti del progetto è la grafica sulla quale non sono riuscito a soffermarmi come avrei desiderato. Ciò nonostante grazie all'aiuto di Bootstrap, raccolta gratuita di modelli tipografici e fogli di stile per applicazioni web, sono riuscito, credo, a dare un bell'aspetto all'interfaccia.

Guardandomi indietro, gli inconvenienti incontrati potevano, in realtà, essere evitati con una più accurata analisi dei rischi nella fase di pre-progetto.

Per concludere, comunque, sono decisamente fiero e contento del lavoro svolto e di come ho portato a termine il progetto.

### Strumenti utilizzati nello svolgimento di questo progetto:

Visual Studio Code - editor di codice sorgente

JDER - Software per la realizzazione del diagramma E/R.

Gantt Project - Software per la gestione di progetti

XAMPP - Pacchetto applicativo contenente un web server, un DBMS, e altri strumenti per la creazione di pagine web dinamiche in PHP

Cisco Packet Tracer - Programma di simulazione di infrastrutture di rete

### Si allegano:

*GusattoDerek\_Achilles.pkt* - file di Packet Tracer in cui è realizzata l'infrastruttura di rete

*GusattoDerek\_Achilles* - cartella contenente i file PHP dell'applicazione

*GusattoDerek\_Achilles.sql* - dump del database realizzato