

INDUSTRIAL PROCESS MANAGEMENT

Il lavoro assegnatoci riguarda la progettazione di un'applicazione per la pianificazione delle azioni di manutenzione all'interno di un ambiente industriale.

La progettazione dell'applicazione si suddivide in diversi step tra cui:

- Scelta dell'architettura software;
- Stesura delle User story;
- Sprint backlog + Prima release;
- Sprint backlog + Seconda release;
- Testing.

Verranno descritte in seguito nel dettaglio le procedure seguite nella progettazione dell'applicazione.

Scelta dell'architettura Software

Il primo step per la progettazione è stata la scelta dell'architettura software. Per lo sviluppo dell'applicazione è stato scelto un tipo di architettura web-based poiché risulta essere facilmente accessibile, più sicura sia a livello software che a livello Hardware, facilmente gestibile e distribuibile agli utenti che dovranno utilizzarla. Essa, inoltre, risulta essere più scalabile e modulare.

L'architettura software da noi scelta è strutturata su tre livelli: Client, Server e Database.



Il Client rappresenta ciò che viene chiamato “Front-End”, ovvero la parte statica dell'applicazione web in cui viene implementato tutto ciò che riguarda l'interfacciamento con l'utente e lo scambio di messaggi tramite il protocollo HTTP. Esso, quindi, effettua delle richieste di dati al server e, una volta ottenuti, li rende disponibili all'utente tramite l'User Interface.

Il Server e il Database, invece, racchiudono ciò che invece riguarda la parte di “Back-End” e quindi la parte dinamica dell'applicazione web. Il server, quindi, elabora la richiesta del client, interroga il Database (se necessario) e costruisce la risposta per il client servendosi degli eventuali dati prelevati.

I linguaggi utilizzati per l'implementazione del Front-End sono:

- L'HTML per la costruzione dello scheletro delle pagine web;
- Il CSS per la rendere la pagina web navigabile e user friendly;
- JavaScript e in particolare il Framework jQuery per lo scambio dei messaggi e le richieste dei dati al server;

mentre quelli utilizzati per il Back-End sono:

- PHP per l'elaborazione delle richieste dei dati da parte del client;
- MySQL per l'implementazione del Database.

Complessivamente, l'architettura si presenta strutturata in modo che ogni livello comunica sia con il livello sottostante che con quello sovrastante.

Distribuzione delle attività e stima dei tempi di progettazione

Per una migliore gestione e cura dei dettagli è stato suddiviso il lavoro tra i vari membri del gruppo. Inizialmente, tutti i membri del gruppo si sono riuniti per decidere delle linee comuni da utilizzare all'interno del codice prodotto e si sono definite delle regole base da seguire durante la progettazione:

Protocollo gruppo:

DESIGN:

- Codice pulito, essenziale e conforme interamente alle richieste
- Dipendenze minime tra le classi, evitando quelle inutili
- Massima riusabilità del codice rendendolo più generico possibile per future implementazioni
- Utilizzare almeno 2/3 Design Patterns
- Non inserire duplicazioni e ripetizioni

CODE:

- Niente abbreviazioni (se un'abbreviazione non è intuibile, si scrive per intero es. variabili)
- Per le classi di test nominarle agganciando il suffisso "tester" (Classe e variabili si parte con la lettera maiuscola, nelle funzioni si parte con la lettera minuscole, costanti tutte maiuscole: **notazione a cammello**)
- Formattazione e indentazione, codice organizzato bene, commenti su più righe, commenti esplicativi, nomi delle funzioni che riflettono bene quello che fanno,
- Evitare hard coding
- Struttura codice leggibile e funzioni che fanno poche cose ma buone

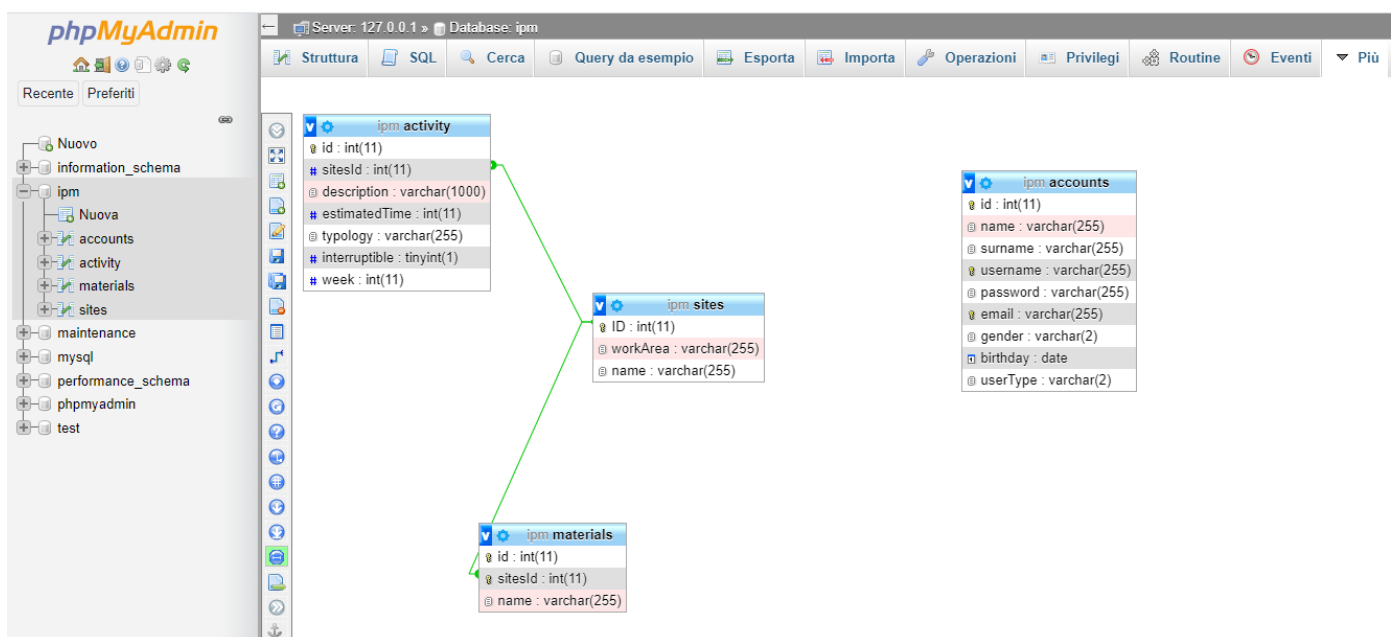
TESTING: l'operazione di testing deve essere fatta su tutti i metodi pubblici delle classi definendo dei **TEST CASE**.

Successivamente si è passato alla stesura delle User Story e alla suddivisione del lavoro mediante degli incarichi assegnati ad ogni membro del gruppo.

Progettazione del Database

Una delle cose fondamentali per la progettazione dell'applicazione è stata la creazione del Database in cui verranno conservati i dati in modo da renderli sempre disponibili al client (Utente) qualora questo ne avesse bisogno. In particolare, è stato utilizzato XAMPP, una piattaforma software costituita dal web server Apache HTTP, il database MySQL e tutti gli strumenti necessari per utilizzare il linguaggio di programmazione PHP.

Strutturalmente esso si presenta strutturato nel seguente modo:



Come è possibile vedere in figura le tabelle sono strutturate in modo che ci fosse un vincolo di integrità referenziale per poter mantenere la consistenza in seguito a operazioni di cancellazione e/o aggiornamento delle tabelle.

Al passo successivo si è provveduto al popolamento del Database inserendo le tabelle e aggiungendo ad esse alcune righe a titolo di esempio. Tra le tabelle abbiamo:

Activity, che gestisce tutte le attività pianificate da un planner ed è legata alla tabella sites attraverso l'ID dell'area di intervento. Ogni attività, quindi avrà associato il nome del sito di intervento e la relativa workArea.

id	sitesId	description	estimatedTime	typology	interruptible	week
1	1	Minor electrical machine problem	90	Electronics	1	52
2	1	Some machines seems to be stuck	30	Mechanical	0	50

Materials, che gestisce la lista dei materiali che sono necessari per le attività di manutenzione in una determinata area infatti anch'essa, così come activity, ha un legame con la tabella sites.

id	sitesId	name
1	1	Tessuto A01

Sites, invece, gestisce le aree in cui è possibile effettuare attività di manutenzione.

id	workArea	name
1	Pellame	Solofra
2	Ingegneria	Scampitella

Accounts, infine, gestisce gli utenti che vengono registrati al sito per poter effettuare correttamente l'operazione di login.

id	name	surname	username	password	email	gender	birthday	userType
1	Alex	Solimine	alessio	12345	alex@solimine.com	M	1996-05-10	SA
2	Emidio	Pierro	emidio	12345	emidio@pierro.com	M	1997-07-21	RM
3	Toni	Pannese	toni	12345	toni@pannese.com	NS	2021-02-10	PL
4	Veronica	Salvati	veronica	12345	veronica@salvati.com	F	2020-12-01	MT

Implementazione di prime pagine HTML e relative funzionalità

Un altro step fondamentale nello sviluppo dell'applicazione web è stata l'implementazione delle prime pagine che andranno a costituire il sito. Alcune di quelle presentate nella prima release, rappresentano solo una bozza poiché sono state implementate semplicemente per avere una prima idea d'insieme, altre invece sono state migliorate.

Per prima cosa ci siamo curati della creazione della struttura HTML delle varie pagine per definire uno scheletro base su cui lavorare. Successivamente le pagine HTML sono state arricchite con dei fogli di stile, implementati con tecnologia CSS, appositamente progettati per rendere unitario il tema dell'applicazione.

È stata scelta una linea comune per tutte le pagine web in modo da rendere l'applicazione facilmente navigabile e interattiva (User Friendly).

Per facilitare la scrittura del codice CSS è stato utilizzato il Framework Bootstrap che contiene modelli di progettazione basati su HTML e CSS.

Infine per l'implementazione della dinamica del sito sono stati implementati degli script in PHP e Javascript che rappresentano, ognuno una funzionalità diversa svolta dalla rispettiva pagina web statica a cui è collegato. In seguito verranno mostrate alcune delle pagine implementate.

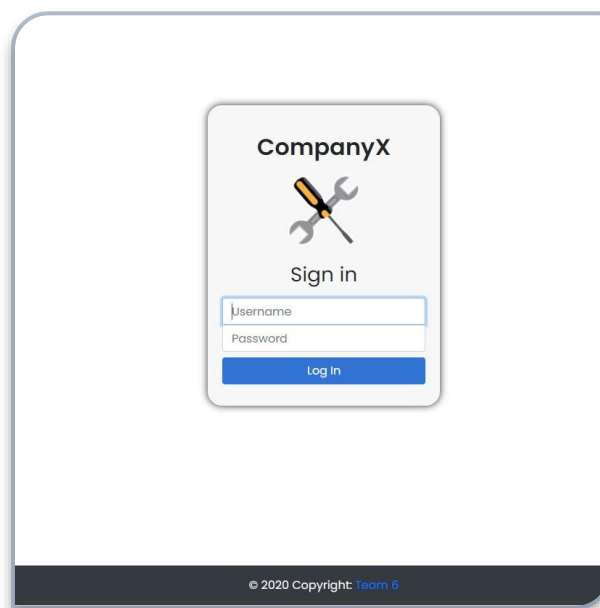
Login page

Come abbiamo detto prima, la pagina di **login** gestisce l'accesso degli utenti registrati al sito. Abbiamo scelto di implementare dapprima questa funzionalità poiché è stato ritenuto opportuno garantire un accesso mutuamente esclusivo per ogni utente e perché è la prima pagina che viene mostrata nel momento in cui si ha accesso al sito.

Quando viene mostrata questa pagina, l'utente deve inserire l'username e la password. Una volta fatto ciò, viene interrogato il database per controllare se i valori inseriti nei campi da compilare matchano con una delle righe presenti nella tabella account all'interno del Database. Se ciò avviene vuol dire che l'utente inserito è registrato al sito e quindi si può procedere al login.

A seconda del tipo di utente che effettua l'accesso, si viene indirizzati nella home page relativa alla mansione dell'utente: Planner, Repository Manager, System Administrator oppure Maintainer.

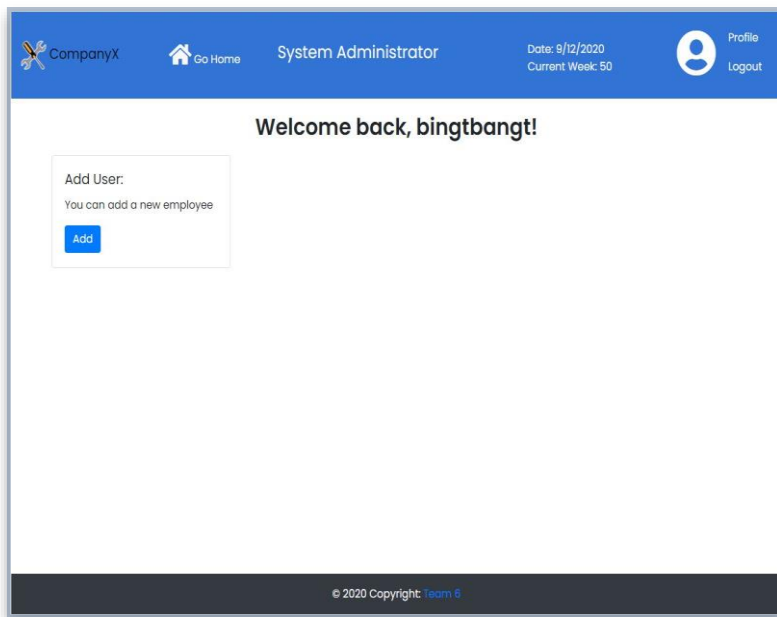
Se ciò non accade viene mostrato il messaggio di errore "username o password non corretti" costringendo l'utente a reinserirli nuovamente.



La scelta di tale messaggio di errore è dovuta a un fatto di sicurezza poiché chiunque tenti di accedere illegalmente al profilo di un utente non sa se ha inserito erroneamente l'username, la password o entrambi.

System Administrator home page

Il **System Administrator** è l'unico ruolo che ha i permessi per quanto riguarda la gestione della registrazione di un nuovo utente all'interno del sito per cui, il primo utente a dover essere inserito all'interno del database, è appunto un SA. Questo inserimento deve essere fatto manualmente altrimenti chiunque, anche chi non è un dipendente diretto dell'azienda può registrarsi. Una volta inserito un SA, egli può aggiungere, eliminare o modificare un utente tramite il pulsante <<Add>>.



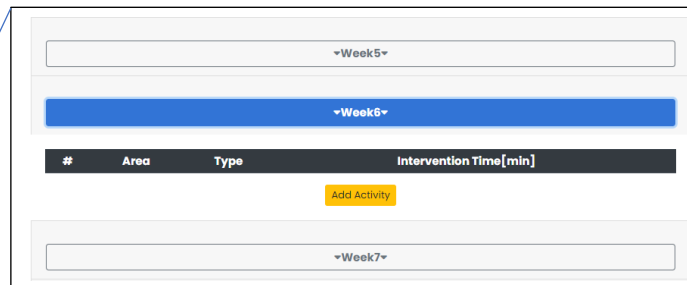
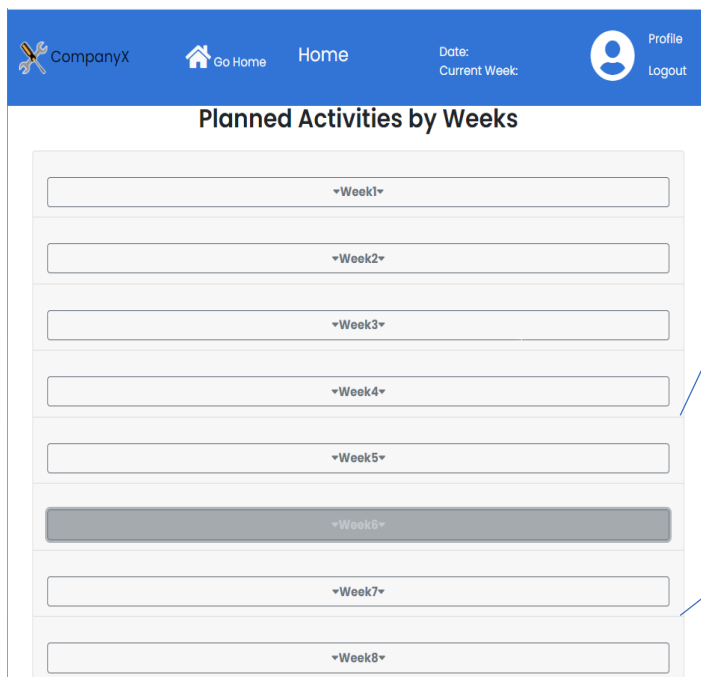
Cliccando su di esso, si accede ad un'ulteriore pagina che contiene il modulo di registrazione di un nuovo utente. Compilando i campi obbligatori e facendo click sul pulsante **«Add User»**, i dati vengono inviati al Database e salvati all'interno di esso.

The screenshot shows a 'New user registration form' within the System Administrator interface. The top navigation bar is blue and contains: a wrench icon and 'CompanyX', a house icon and 'Go Home', the text 'SA Add New User', the date 'Date: 9/12/2020' and 'Current Week: 50', and a user profile icon with 'Profile' and 'Logout' links. The form itself is a light grey box with the title 'New user registration form'. It contains the following fields: 'Name' and 'Surname' (text inputs), 'Username' and 'Password' (text inputs), 'Email' (text input), 'Gender' (a dropdown menu with 'Male' selected), 'Date of Birth' (a date picker showing 'gg/mm/aaaa'), and a role selection section with three radio buttons: 'System Administrator', 'Repository Manager', and 'Planner' (which is selected), and a 'Maintainer' option with a blue dot. A blue 'Add User' button is at the bottom of the form. The footer is dark grey with the text '© 2020 Copyright: Team 6'.

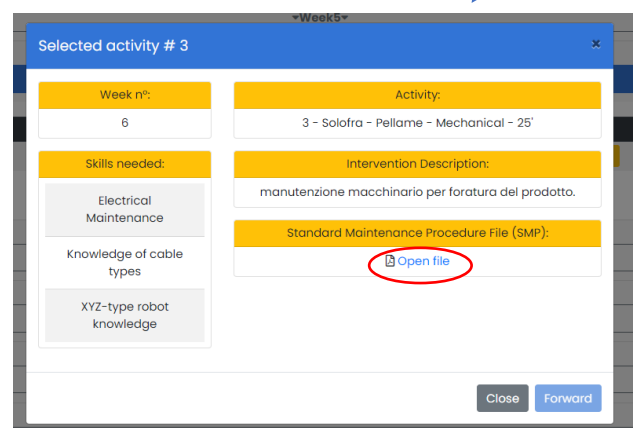
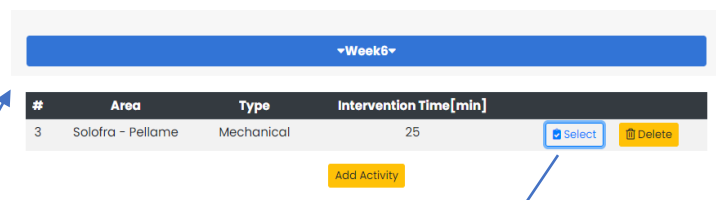
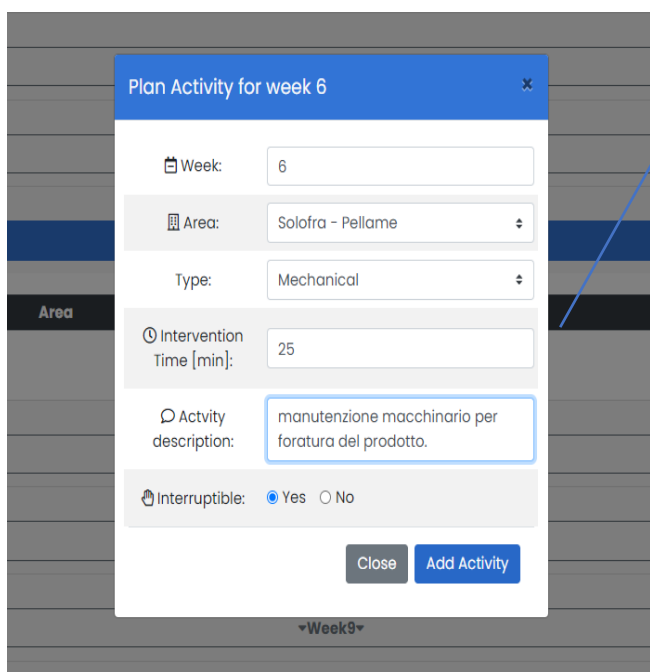
Infine, cliccando sull'icona "Go Home" in alto a sinistra si può ritornare alla relativa schermata di home.

[Planner home page](#)

Il planner è un'utente che può pianificare le attività dei vari manutentori per settimana. Vediamo che la pagina è stata strutturata in modo che, scorrendo verso il basso, vi siano le varie settimane e cliccando su ognuna di esse viene aperto un menu in cui è possibile aggiungere un'attività.



cliccando sul pulsante **“Add Activity”** è possibile aggiungere una nuova attività inserendo l’area di interesse, il tipo di intervento, il tempo stimato, una breve descrizione e specificando se tale attività può essere o meno interrotta. Una volta aggiunta, questa verrà visualizzata all’interno della lista delle attività pianificate. Il planner può visualizzare le attività pianificate cliccando su **“Select”** e scaricare un file PDF relativo alle procedure standard di manutenzione (SMP). Infine, può eliminare un’attività facendo click sul pulsante **“Delete”**.



Maintainer home page

Il **Maintainer** può visualizzare tutte le attività che sono gli sono state assegnate. Egli deve poter rendere disponibile al planner un file PDF che racchiude una timeline riguardante la propria disponibilità per eventuali attività di manutenzione improvvise. Il maintainer, inoltre, deve poter avere accesso al file PDF

messo a disposizione dal Planner che contiene le procedure standard relative all'attività di manutenzione (SMP) per poter svolgere al meglio il lavoro che gli è stato attribuito.

The screenshot shows the 'Maintainer Home Page' interface. At the top, there's a blue header with 'CompanyX', 'Go Home', 'Maintainer Home Page', and user options like 'Date: Current Week', 'Profile', and 'Logout'. Below the header, a 'Welcome back!' message is followed by a table of activities. The table has columns: ID, Area, Tipology, and Estimate Intervention Time. One activity is listed with ID 24332, Area Fisciano, Tipology Electric, and Estimate 123. A 'Select' button is next to it. Below the table, there are input fields for 'Area', 'Choose Type', 'Est. Time', and buttons for 'Submit' and 'Delete'. A modal window titled 'Selected activity #' is open, showing fields for 'Week n°', 'Activity:', 'Workspace Notes:', 'Intervention Description:', 'Skills needed:', 'Standard Maintenance Procedure File (SMP):' (with an 'Open file' link), and 'Maintener Availability:' (with an 'Open file' link). The modal has 'Close' and 'Forward' buttons at the bottom.

Repository Manager home page

Il **Repository Manager** può gestire i siti e le work Area in cui è prevista l'attività di manutenzione: Egli può aggiungere, cancellare o modificare un sito di intervento.

Manage Sites

id	Site	Work Area	
1	Solofra	Pellame	Delete
2	Scampitella	Ingegneria	Delete

[Add](#)

Il RM può, inoltre gestire la lista dei materiali utili per una determinata attività di manutenzione: egli può aggiungere, eliminare e modificare un materiale all'interno della lista.

Manage Materials

id	Site	Material	
1	Solofra - Pellame	Tessuto A01	Delete

[Add](#)

Profile

Nelle home page di tutti gli utenti è presente una sezione nella quale è possibile visualizzare tutti i dati del profilo utente che sono stati inseriti durante la procedura di registrazione. Il profilo di ogni utente può essere eventualmente aggiornato tramite il pulsante «Update».

The screenshot shows a web application interface. At the top is a blue navigation bar with the following elements from left to right: a logo for 'CompanyX', a 'Go Home' button with a house icon, the text 'Profile Page', the date 'Date: 9/12/2020' and 'Current Week: 50', and a user profile icon with 'Profile' and 'Logout' links. Below the navigation bar is a white profile card with a grey border. The card is titled 'Profile' and contains the following user information: Name: Veronica, Surname: Salvati, Email: veroska@hotmail.it, Gender: F, Date of Birth: 1996-09-24, and User Type: PL. At the bottom of the card is a yellow 'Update' button. Below the profile card is a dark grey footer bar with the text '© 2020 Copyright: Team 6'.

Infine eseguendo il logout è possibile uscire dalla propria area, previa conferma da parte dell'utente stesso. Eseguendo il logout si viene reindirizzati direttamente alla pagina di login iniziale.

Testing

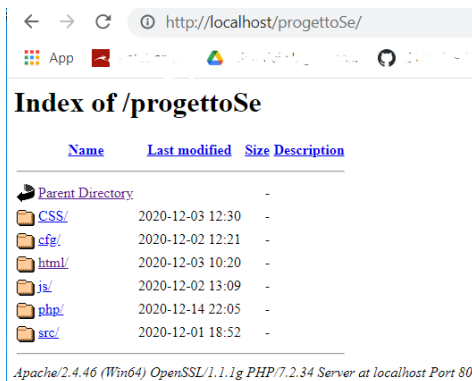
Per quanto riguarda il testing è stato utilizzato un approccio di tipo **“white-box Testing”**. In questo tipo di approccio, i test sono basati sulla conoscenza della struttura del programma e dei suoi componenti. Per poter approcciare a questo tipo di testing è importante poter avere accesso al codice sorgente dell'applicazione. In questo metodo, la conoscenza del codice sorgente dell'applicazione viene utilizzata per creare dei test di “difetto” e l'obiettivo è quello di progettare casi di test che forniscano un certo livello di copertura del programma e garantiscano che queste funzioni correttamente.

Per fare ciò, quindi, è stato necessario produrre dapprima le funzioni e, man mano, queste sono state testate per verificare il corretto funzionamento di ognuna di esse.

Per l'attività di testing è stata usata anche la tecnica del testing di regressione che è una tipologia di [test del software](#) ha lo scopo di verificare la qualità di nuove versioni di un prodotto. L'intento è quello di testare le nuove funzionalità e garantire che le funzionalità preesistenti abbiano mantenuto le loro caratteristiche qualitative dopo l'introduzione di queste ultime.

Poiché l'implementazione dell'applicazione software è stata implementata seguendo una strategia di tipo bottom-up ogni singola componente sviluppata veniva testata per controllarne il corretto funzionamento e ogni volta che veniva implementata una nuova funzionalità si controllava se la qualità del prodotto fosse rimasta tale, nel senso che abbiamo verificato che le funzionalità precedentemente testate continuassero a funzionare correttamente anche dopo aver apportato le modifiche all'applicativo software.

Il testing è stato fatto tramite la piattaforma software XAMPP con la quale, una volta avviati il server Apache e il Database MySQL dal pannello di controllo, si inserisce l'indirizzo <http://localhost/progettoSE> all'interno del browser web e, a questo punto, comparirà la home page del server virtuale in cui ci saranno tutte le cartelle relative all'applicazione sviluppata.

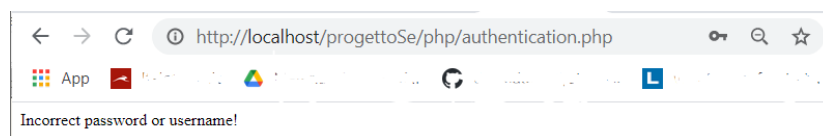


Tramite questo metodo è possibile testare interattivamente il comportamento del sito web confrontandolo con il comportamento desiderato del sistema. Per fare in modo che questo tipo di testing funzioni correttamente è bene che l'autore dell'applicativo abbia le idee ben chiare su ciò che si aspetta di vedere durante l'esecuzione.

Sono state testate due delle funzionalità ritenute di particolare importanza per l'applicazione. Queste sono:

- 1) Il meccanismo di autenticazione per rendere esclusivo l'accesso ai singoli utenti;
- 2) La persistenza della connessione.

Per testare il meccanismo di autenticazione è stato inserito un utente come System Administrator all'interno della tabella **account** nel Database creato in con PhpMyAdmin. Successivamente all'indirizzo <http://localhost/ProgettoSE/html> viene visualizzata la pagina di login. Inserendo l'username e la password nei campi compilabili, si può vedere che se la richiesta di login ha successo la pagina viene aggiornata con quella relativa all'utente loggato. Se ciò non accade, inserendo i dati di un utente non presente nel Database si può vedere un messaggio di errore di questo tipo:



Come è già stato detto, si è scelto di inserire questo messaggio di errore così che l'utente non potesse sapere se è stato inserito erroneamente soltanto l'username o soltanto la password (Security).

Infine è stata testata anche la persistenza della connessione ovvero che una volta che un utente ha fatto l'accesso al proprio account, dopo un certo tempo di inattività del sito, la connessione non viene persa e quindi si evita di rifare il login.

Grazie all'interattività di questo metodo di testing è stato possibile scoprire questo piccolo bug del sistema poiché sono state fatte prove di funzionamento in tempo reale e ci si è accorti della mancanza della funzione che garantisce il mantenimento della sessione. La mancanza, quindi, della funzione `session_start()` non consentiva di salvare le variabili di sessione e quindi dopo un certo tempo l'applicazione ritornava sempre sulla pagina di login.

Team 6

Alessio Solimine

Emiddio Pierro

Toni Pannese

Veronica Salvati