Gestione progetti aziendali

Sommario

[Analisi dei requisiti 3](#_Toc154050527)

[Situazione iniziale 3](#_Toc154050528)

[Spiegazione elementi tabella dei requisiti: 3](#_Toc154050529)

[Metodologia lavoro 4](#_Toc154050530)

[Architettura del programma 5](#_Toc154050531)

[Er 5](#_Toc154050532)

[Conclusioni 6](#_Toc154050533)

[Mancanze conosciute 6](#_Toc154050534)

[Identificativo 6](#_Toc154050535)

[Gestione solo dei propri progetti admin 6](#_Toc154050536)

[Personali 6](#_Toc154050537)

# Analisi dei requisiti

## Situazione iniziale

Siamo stati incaricati da una azienda familiare per la creazione di un loro sito interno per la gestione dei progetti. Questa applicazione aiuterà i dipendenti a gestire tutti i progetti in modo veloce, efficiente e sicuro come la possibilità di attribuire i vari progetti solamente ai collaboratori presenti nell’azienda. Queste feature rendono molto più semplice e ordinata la gestione dei lavori per i dipendenti.

## Spiegazione elementi tabella dei requisiti:

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

Nelle tabelle sottostanti si potranno visualizzare i requisiti ricavanti dalle richieste del committente:

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Applicazione web in PHP |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di un’applicazione web creata con PHP |
| **Sotto requisiti** | |
| **001** | Bisogna utilizzare l’MVC di Sartori |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | History |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma dovrà avere una history contenente tutti i progetti tenendo traccia dei loro relativi stati di avanzamento e il collaboratore che se ne sta occupando. |
| **Sotto requisiti** | |
| **001** | I progetti terminati dovranno essere mostrati nell’archivio (interfaccia separata) e non potranno essere modificati o eliminati. |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Creazione progetti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il programma permetterà la creazione dei nuovi progetti. |
| **Sotto requisiti** | |
| **001** | Per creare i progetti bisognerà inserire un titolo, un codice identificativo del progetto, una descrizione, la dati di inizio e l’assegnazione del collaboratore. |
| **002** | L’Identificativo dovrà iniziare con le ultime cifre dell’anno corrente e con un numero incrementale a 3 cifre. Esempio: 23001, 23002, 23003, … |
| **003** | Il programma dovrà tener traccia dello stato di ogni progetto e ad ogni suo cambiamento verrà memorizzata la data. |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Ricerca |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | È possibile filtrare i progetti per nome |

# Metodologia lavoro

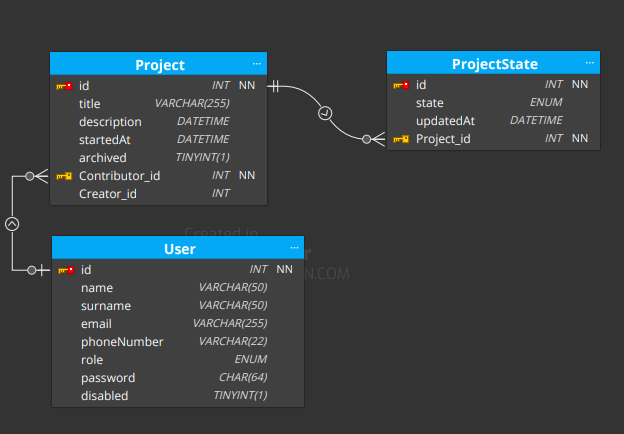
Dal momento che questo lavoro doveva essere svolto a coppie ci siamo divisi i compiti in questo modo:

* **Siro** ha gestito tutta la parte dei progetti e tutta la grafica delle pagine rendendo l’applicativo bello esteticamente.
* **Alessandro** ha lavorato sulla gestione degli utenti e del login potendo così gestire i permessi in modo ottimale.

Ci sono state delle parti dove abbiamo lavorato assieme come per la documentazione o la gestione dei permessi dove risultava molto più efficiente inserire i controlli alle proprie pagine.

# Architettura del programma

## Er



L’immagine sovrastante rappresenta il diagramma Entità-Relazione dell’applicativo.

La tabella dei progetti possiede:

* **id**: identificativo del progetto (formato: AAXXX 🡪 ultime 2 cifre dell’anno + numero a 3 cifre)
* **title**: titolo del progetto
* **description**: descrizione del progetto
* **startedAt**: data di creazione di progetto
* **Contributor\_id**: id del collaboratore del progetto

La tabella degli utenti possiede:

* **id**: identificativo dell’utente
* **name**: nome dell’utente
* **surname**: cognome dell’utente
* **email**: email dell’utente
* **phoneNumber**: numero di telefono dell’utente (max. 22 caratteri, senza spazi)
* **password**: password dell’utente (SHA256 encryption)

La tabella degli stati possiede:

* **id**: identificativo dello stato
* **state**: stato del progetto (Assegnato, In corso, test, concluso)
* **updatedAt**: data di modifica dello stato
* **Project\_id**: id del progetto di riferimento

# Conclusioni

Giunti alla fine del progetto abbiamo sviluppato un applicativo quasi completo, le mancanze sono poche e le funzioni più importanti sono tutte implementate.

## Mancanze conosciute

### Identificativo

A causa del tempo limitato e degli imprevisti riscontrati durante l’implementazione non siamo riusciti a implementare questa feature. Gli id non hanno nessun formato e sono inseriti in modo univoco senza tener conto della data.

### Gestione solo dei propri progetti admin

Gli admin possono modificare ed eliminare tutti i progetti e non solo i propri, questa mancanza è data da una dimenticanza di una colonna nel db che indicasse chi ha creato il progetto. Dal momento che ci siamo accorti tardi di questa dimenticanza, abbiamo implementato solo in parte questa funzione inerendo all’interno del database un campo che non viene utilizzato.

## Personali

**Alessandro**:

Dal momento che lo scorso anno non sono riuscito bene a capire php grazie a questo progetto sono riuscito a mettere assieme tutte le competenze imparate nelle scorse lezioni riuscendo a capire molto di più come gestire dati, utenti e l’mvc. Sono riuscito a mettere in pratica in modo efficiente anche le competenze acquisite quest’anno come il PDO e le Prepared-Statement riuscendo a comprenderle in modo ottimale.

**Siro:**

Con questo progetto ho potuto imparare diverse cose che gli anni scorsi non ero riuscito a comprendere, come ad esempio la gestione delle sessioni con PHP.

Generalmente sono soddisfatto del risultato anche se non è finito al 100%.