

# BalanceBuddy Project

Tommaso Polvere  
Alberto Coscetti  
Rayan Marzouki

Marzo 2024

## Indice

<b>1</b>	<b>Analisi dei Requisiti</b>	<b>3</b>
1.1	Introduzione . . . . .	3
1.1.1	Scopo . . . . .	3
1.1.2	Scopo del progetto . . . . .	3
1.1.3	Riferimenti . . . . .	3
1.1.4	Panoramica del documento . . . . .	3
1.2	Descrizione Generale . . . . .	3
1.2.1	Prospettive del prodotto . . . . .	3
1.2.2	Funzionalità del prodotto . . . . .	3
1.2.3	Classificazione degli utenti . . . . .	3
1.2.4	Vincoli generali . . . . .	4
1.2.5	Assunzioni e Dipendenze . . . . .	4
1.3	Requisiti Specifici . . . . .	4
1.3.1	Interfaccia utente . . . . .	4
1.3.2	Interfaccia hardware . . . . .	4
1.3.3	Interfaccia software . . . . .	4
1.3.4	Interfaccia di comunicazione . . . . .	4
1.4	Requisiti funzionali . . . . .	4
1.4.1	Inserimento di un nuovo campo spesa . . . . .	4
1.4.2	Visualizzazione delle spese . . . . .	4
1.4.3	Calcolo della spesa media mensile . . . . .	5
1.4.4	Impostazione del budget mensile . . . . .	5
1.5	Requisiti non funzionali . . . . .	5
1.5.1	Sicurezza . . . . .	5
1.5.2	Performance e Scalabilità . . . . .	5
1.5.3	Manutenzione . . . . .	5
1.6	Requisiti di dominio . . . . .	5
1.6.1	Autenticazione . . . . .	5

<b>2</b>	<b>Suddivisione dei compiti</b>	<b>5</b>
2.1	Introduzione . . . . .	5
2.2	Front-end Development . . . . .	6
2.3	Back-end Development . . . . .	6
<b>3</b>	<b>Tecnologie utilizzate</b>	<b>6</b>
3.1	Tecnologie Front-end . . . . .	6
3.2	Tecnologie Back-end . . . . .	6
3.3	Altre tecnologie . . . . .	6
<b>4</b>	<b>Versioning e Repository</b>	<b>6</b>
4.1	Versioning . . . . .	6
4.2	Deployment finale . . . . .	7
<b>5</b>	<b>Diagramma di GANTT</b>	<b>7</b>
<b>6</b>	<b>Piano dei test</b>	<b>7</b>
<b>7</b>	<b>Conclusione</b>	<b>7</b>
<b>8</b>	<b>Realizzazione Database</b>	<b>8</b>
8.1	Progettazione Concettuale . . . . .	8

# 1 Analisi dei Requisiti

## 1.1 Introduzione

### 1.1.1 Scopo

Lo scopo di questo progetto è quello di realizzare un'applicazione Web che aiuti gli utenti nella gestione delle proprie spese personali. Questo documento spiegherà lo scopo e le caratteristiche del sistema, le interfacce del sistema, cosa farà il sistema, i vincoli in base ai quali deve operare.

### 1.1.2 Scopo del progetto

Questa applicazione web sarà in grado di calcolare le spese medie fatte dall'utente, impostare budget massimi e notificare il fatto di averlo superato o meno. Inoltre l'applicazione deve poter far aggiungere all'utente nuove spese effettuate e ordinarle (*categoria, prezzo, data di inserimento...*) .

### 1.1.3 Riferimenti

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

### 1.1.4 Panoramica del documento

La restante parte di questo documento contiene una descrizione dettagliata e approfondita delle funzionalità richieste al sistema.

## 1.2 Descrizione Generale

### 1.2.1 Prospettive del prodotto

L'applicazione non è parte e non deve integrarsi in alcun sistema software o hardware, essendo un' applicazione completamente autonomo.

### 1.2.2 Funzionalità del prodotto

L'applicazione web deve:

- Aggiungere nuove spese personali.
- Ordinarle e/o filtrarle per categoria, data di inserimento, prezzo.
- Impostare budget massimi mensili.
- Calcolare l'importo medio spese mensilmente.

### 1.2.3 Classificazione degli utenti

L'app è rivolta ad una utenza con discreta conoscenza dell'applicativo ma senza particolari conoscenze informatiche. All'utente è richiesta solamente una conoscenza informatica di base.

#### 1.2.4 Vincoli generali

Le limitazioni e/o vincoli che possono essere riscontrate durante la fase di sviluppo potrebbero esser causati a difficoltà di utilizzi di strumenti per implementare l'applicativo.

#### 1.2.5 Assunzioni e Dipendenze

L'applicazione deve poter girare su un **Server Web** sul quale fare il *deployment* dell'app; deve quindi esser accessibile mediante browser.

### 1.3 Requisiti Specifici

#### 1.3.1 Interfaccia utente

L'applicazione web deve esser dotato di pagine *HTML* composte da pulsanti, campi testuali dove scrivere.

#### 1.3.2 Interfaccia hardware

L'applicazione non si deve interfacciare con nessun sistema hardware.

#### 1.3.3 Interfaccia software

L'applicazione non si deve interfacciare con nessun sistema software.

#### 1.3.4 Interfaccia di comunicazione

L'applicazione non deve interfacciarsi con nessun protocollo di comunicazione.

### 1.4 Requisiti funzionali

In questa sezione vengono trattate tutte le funzioni principali che deve eseguire l'applicativo web.

#### 1.4.1 Inserimento di un nuovo campo spesa

- **Introduzione:** Consente di inserire un nuovo campo spesa nel sistema.
- **Input:** Importo spesa, categoria, data.
- **Elaborazione:** L'applicazione prende le varie informazioni inserite e aggiunge una nuova *tupla* nel *DB(DataBase)* per memorizzare le nuove informazioni.
- **Output:** Restituisce un messaggio di conferma/errore dell'inserimento.

#### 1.4.2 Visualizzazione delle spese

- **Introduzione:** Consente di visualizzare le spese inserite nel sistema.
- **Input:** Nessun dato.
- **Elaborazione:** L'applicazione stampa a schermo le spese inserite
- **Output:** Visualizzazione delle spese già precedentemente inserite.

#### 1.4.3 Calcolo della spesa media mensile

- **Introduzione:** Consente di visualizzare la spesa media per ogni mese.
- **Input:** Nessun dato.
- **Elaborazione:** L'applicazione calcola la media dei dati raccolti dal *DB(DataBase)*.
- **Output:** Visualizzazione della media calcolata.

#### 1.4.4 Impostazione del budget mensile

- **Introduzione:** Consente di impostare il budget massimo da poter spendere per ogni mese.
- **Input:** Budget massimo.
- **Elaborazione:** L'applicazione controlla se le spese effettuate hanno "sforato" il budget massimo.
- **Output:** Notifica di "sforamento" del budget (se superato).

### 1.5 Requisiti non funzionali

In questa sezione sono riportati i requisiti non funzionali necessari per il corretto funzionamento dell'applicativo web.

#### 1.5.1 Sicurezza

Specifica i fattori che proteggeranno il software da accessi accidentali o dannosi, uso improprio o modifica.

#### 1.5.2 Performance e Scalabilità

Specifica i fattori che non rendono il software lento rispetto al numero delle richieste utente.

#### 1.5.3 Manutenzione

Specifica i fattori che rendono il software facile da sistemare o aggiornare.

### 1.6 Requisiti di dominio

#### 1.6.1 Autenticazione

L'applicazione deve consentire all'utente di registrarsi ed autenticarsi per accedere alla piattaforma.

## 2 Suddivisione dei compiti

### 2.1 Introduzione

In questa sezione vengono approfonditi i dettagli riguardo la suddivisione dei compiti all'interno del *gruppo di lavoro*.

## 2.2 Front-end Development

Il *Front-end developer* si occupa di realizzare attraverso linguaggi di *mark-up* come *HTML, CSS, JS*.. pagine web che fungono da interfaccia tra l'utente e tutto il *back-end* (il sistema che sta dietro e che permette di far funzionare l'applicazione). Responsabile: **Rayan Marzouki Alberto Coscetti**.

## 2.3 Back-end Development

Il *Back-end developer* si occupa di realizzare attraverso linguaggi di *programmazione* come *PHP, SQL*.. tutto l'apparato che permette di implementare le varie funzioni dell'applicazione. Responsabile: **Tommaso Polvere**.

# 3 Tecnologie utilizzate

## 3.1 Tecnologie Front-end

Le tecnologie utilizzate per lo sviluppo di pagine web sono linguaggi di *mark-up* e in particolare:

- *HTML*: per la struttura delle pagine.
- *CSS*: per lo stile delle pagine web.
- *JS*: per la dinamicità di alcune funzioni particolari delle pagine web.

## 3.2 Tecnologie Back-end

Le tecnologie utilizzate per lo sviluppo del back-end sono in particolare:

- *PHP*: come tramite tra il front-end e il server Web dove sono memorizzate le informazioni.
- *SQL*: per scrivere le query adeguate per gestire le *tuple* con le informazioni.

## 3.3 Altre tecnologie

Per la realizzazione di tale applicazione sono usati ambienti di sviluppo di tipo *Visual Studio Code, Notepad++* per il front-end. Per la parte di realizzazione di *RDB(Reletion-DataBase)* sono utilizzati ambienti *XAMPP, PHPMyAdmin*.

# 4 Versioning e Repository

## 4.1 Versioning

Per effettuare il *versioning* del progetto, si utilizza il software *Git* per effettuare *commit* di versioni del progetto e il sito web associato *Git-Hub* per accedere al progetto su più *device* contemporaneamente.

Il link alla repository è: <https://github.com/Tommaso1809/BalanceBuddy.git>.

## 4.2 Deployment finale

Per rilasciare il progetto finale l'idea è quella di fare il *deploy* del sito su un *Server Web* tramite siti web che permettono di acquistare spazi su server per caricarci sopra i file.

## 5 Diagramma di GANTT

Qui sotto è riportato il *GANTT CHARTER* dello sviluppo delle varie attività. L'unità di misura è espressa in *giorni* della settimana.

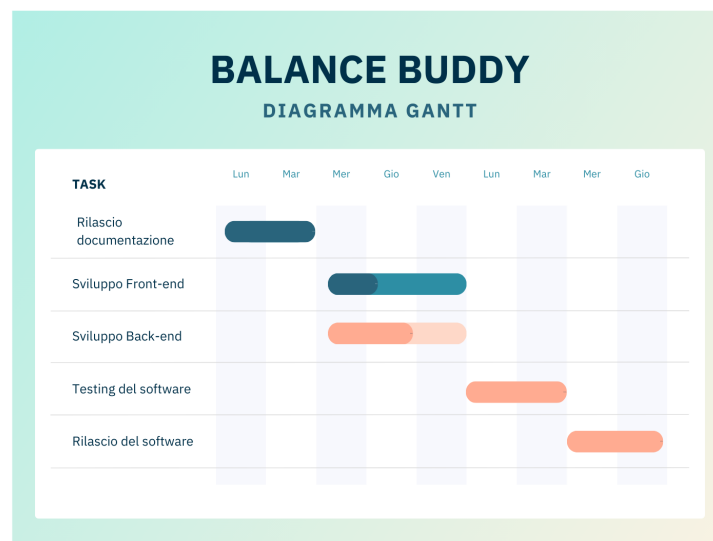


Figura 1: GANTT CHARTER

## 6 Piano dei test

La sezione di *Testing* dell'applicazione verrà effettuata in 2 momenti separati. La **prima** verrà effettuata sul prodotto in *beta Test* come prototipo funzionante in locale sulle proprie macchine. La *seconda* verrà effettuata una volta fatto il *deployment* fisico sul server Web, per verificare la correttezza e coerenza del prodotto realizzato.

## 7 Conclusione

Questo documento è soltanto introduttivo rispetto a quello che il gruppo di lavoro è intenzionato a fare. Eventuali sviluppi e problemi riscontrati verranno approfonditi successivamente una volta rilasciato il prototipo. Lo stesso vale anche per gli schemi (*UML, concettuali, logici, ...*).

## 8 Realizzazione Database

In questa sezione verranno approfonditi i dettagli riguardo la progettazione del DB(DataBase), attraverso le 3 fasi di progettazione (concettuale,logica,fisica).

### 8.1 Progettazione Concettuale

Nell'attività di progettazione e implementazione di una base di dati, una volta superata la prima fase che consiste nella raccolta e analisi dei requisiti la seconda fase consiste nel passaggio da specifiche in linguaggio naturale alla progettazione concettuale del database.

La progettazione concettuale permette di focalizzare l'attenzione sul significato che avranno i vari dati memorizzati nella base di dati, senza scendere in dettagli troppo tecnici che in una prima fase di progetto risultano superflui. Il risultato, quindi,della progettazione concettuale è un diagramma che mette in risalto le entità.

Esistono vari tipi di modello dei dati concettuale che ci permettono di assolvere i compiti sopracitati, quello più diffuso ed anche quello utilizzato in questo contesto è il Diagramma E/R.

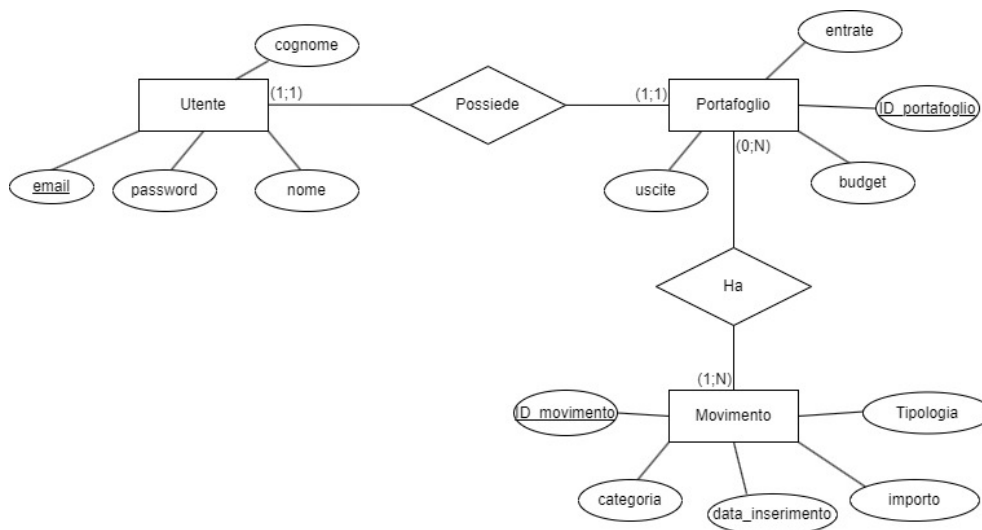


Figura 2: Schema E/R