

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Corso di Laurea Magistrale in Ingegneria Informatica

Classe LM-32

# A Seclab Thesis

A Seclab Subtitle

Relatore

Chiar.mo Prof. John Smith

Tesi di Laurea Magistrale

John Doe

Matricola n. 177013

ANNO ACCADEMICO 2021/2022



## Sommario

Questa tesi si propone di identificare il miglior controllore PID per il controllo della tensione di rete di un islanded microgrid, utilizzando criteri di taratura noti come Ziegler-Nichols e Cohen-Coon ed altri meno conosciuti. Un microgrid, o microrete, è un sistema energetico locale che integra varie fonti di generazione di energia(pannelli solari, turbine eoliche, generatori a combustibile, ecc..) con sistemi di accumulo(batterie) e dispositivi di gestione dell'energia(inverter) per alimentare un'area geografica circoscritta.

Quando una microrete opera in modalità "in isola"(islanded), è disconnessa dalla rete elettrica principale e funziona autonomamente con microsorgenti e carichi propri.

La procedura adottata inizia con la progettazione dello schema elettrico del sistema monofase del microgrid. Successivamente, viene sviluppato un sistema dinamico dal quale si ricava la funzione di trasferimento. Utilizzando quest'ultima, si impiega il software MATLAB/SIMULINK per eseguire la taratura del regolatore PID attraverso algoritmi di autotuning.

I risultati dei test indicano che il PID con le migliori performance sia quello tarato secondo la metodologia di Cohen-Coon, con un tempo di salita di 0.00491 ms, un tempo di picco di 0.0178 ms, un tempo di assestamento di 0.0267 ms ed un overshoot di 2.18 s.



# Indice

<b>1</b>	<b>Iterazione 0</b>	<b>1</b>
1.1	Introduzione . . . . .	1
1.2	Casi d'Uso . . . . .	2
1.3	Priorità casi d'uso . . . . .	4
1.4	Architettura . . . . .	5
1.5	Toolchain . . . . .	7



# Elenco delle figure

1.1	Diagramma casi d'uso . . . . .	3
-----	--------------------------------	---





# Elenco delle tabelle

1.1	Coda alta priorità . . . . .	5
1.2	Coda media priorità . . . . .	6
1.3	Coda media priorità . . . . .	6



# Capitolo 1

## Iterazione 0

### 1.1 Introduzione

**Cos'è Spendly?** Spendly è una web-app progettata per semplificare la gestione delle spese, sia personali che di gruppo. Che si tratti di dividere le spese di una cena tra amici, gestire un budget familiare o monitorare le tue finanze personali, Spendly offre un ambiente intuitivo e funzionalità avanzate per mantenere tutto sotto controllo.

Spendly si distingue grazie all'utilizzo di un algoritmo basato sui minimi cammini Dijkstra, che consente di ottimizzare la distribuzione dei debiti all'interno di un gruppo. Questo significa che, invece di calcoli complessi o confusioni legate ai rimborsi, la web-app genera un sistema chiaro e diretto per regolare i debiti tra i partecipanti, in questo modo, Spendly semplifica la gestione finanziaria, riduce le complessità e promuove la trasparenza.

Spendly è strutturata attorno a una serie di funzionalità che coprono ogni aspetto della gestione delle spese. Queste funzionalità sono state sviluppate con l'obiettivo di offrire la massima flessibilità, sia per gli utenti individuali che per i gruppi di condivisione spese

**Perché scegliere Spendly?** Spendly è più di una semplice applicazione di gestio-

ne delle spese. È una soluzione completa che integra funzionalità avanzate, come il calcolo automatico dei debiti e la gestione centralizzata delle spese. Questo la rende ideale per ogni tipo di utente, dai gruppi di amici che vogliono semplificare la divisione delle spese, alle famiglie che desiderano tenere sotto controllo i propri budget.

Con una piattaforma sicura, flessibile e facile da usare, Spendly trasforma il modo in cui le persone gestiscono i loro soldi, riducendo lo stress finanziario e migliorando la trasparenza nelle relazioni economiche.

Spendly è lo strumento perfetto per chi desidera una gestione semplice, efficace e organizzata delle spese. Con funzionalità pensate per rispondere alle esigenze di utenti individuali e gruppi, questa web-app rappresenta una soluzione innovativa e accessibile per migliorare la vita quotidiana di chiunque voglia un controllo totale sulle proprie finanze.

## 1.2 Casi d'Uso

Di seguito sono riportati alcuni casi d'uso principali:

- **UC1 - Login:** Come utente, voglio poter accedere al mio account per gestire le mie spese.
- **UC2 - Registrazione:** Come nuovo utente, voglio potermi registrare al sistema per iniziare a usare la web-app.
- **UC3 - Logout:** Voglio poter terminare la mia sessione.
- **UC4 - Crea alert di gruppo:** Come utente amministratore di un gruppo spesa voglio poter inserire un alert, dove un alert è un avviso che ci permette di avvisare se si sta raggiungendo una soglia limite di spesa.
- **UC5 - Modifica alert:** Voglio poter modificare i valori dell'alert.
- **UC6 - Elimina alert:** Voglio poter eliminare l'alert.
- **UC7 - Crea gruppo:** Voglio poter creare un gruppo di condivisione spese.

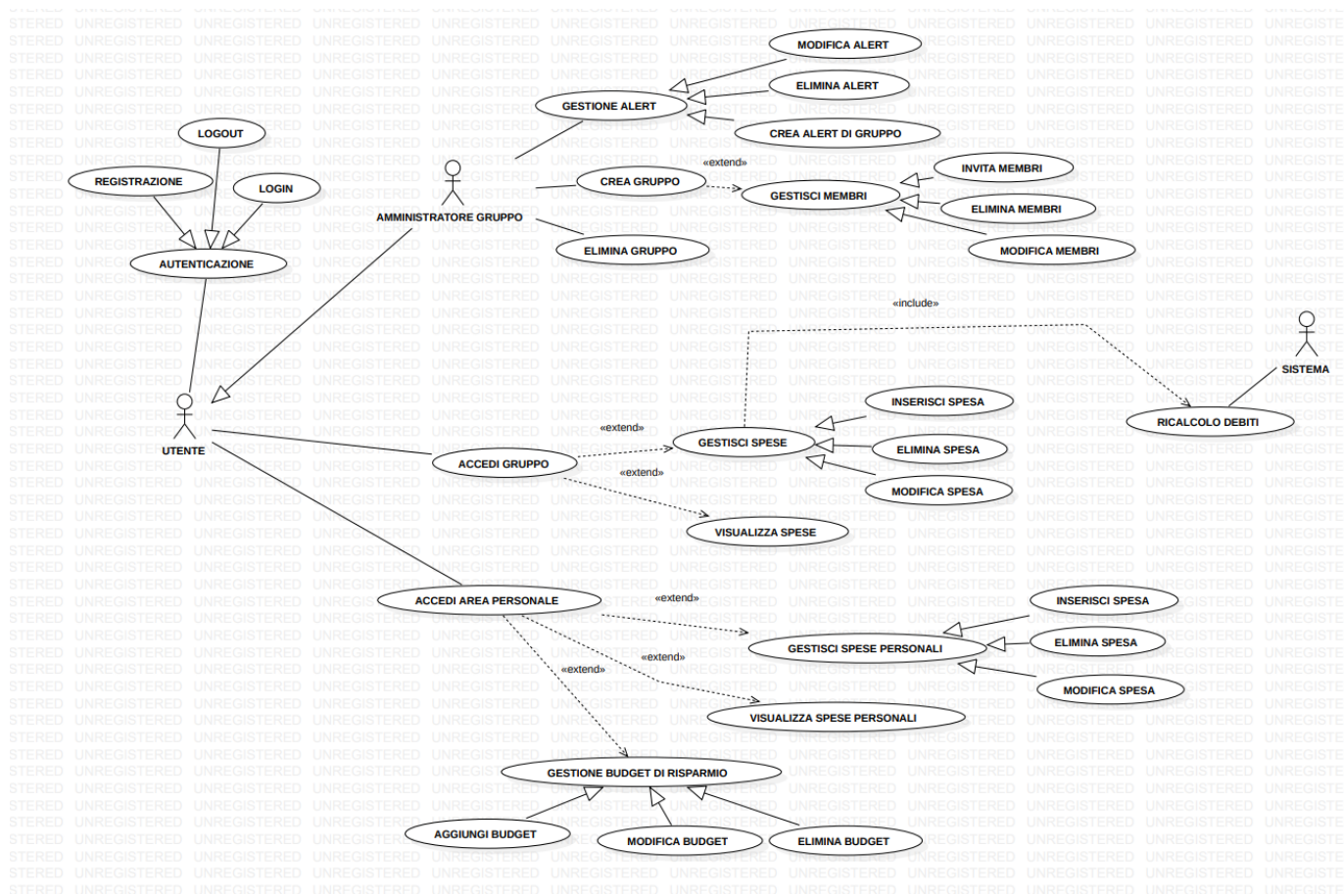


Figura 1.1: Diagramma casi d'uso

- **UC8 - Invita memebri:** Voglio come amministratore invitare membri nel gruppo di spese.
- **UC9 - Elimina membri:** Voglio come amministratore poter eliminare membri del gruppo di spese.
- **UC10 - Modifica membri:** Voglio come amministratore modificare i membri nel gruppo di spese.
- **UC11 - Elimina gruppo:** Voglio poter eliminare un gruppo di condivisione spese.
- **UC12 - Accedi gruppo:** Voglio poter accedere ad un gruppo di condivisione spese.
- **UC13 - Inserisci spesa:** Voglio poter inserire una spesa di gruppo.

- **UC14 - Elimina spesa:** Voglio poter eliminare una spesa di gruppo.
- **UC15 - Modifica spesa:** Voglio poter modificare una spesa di gruppo.
- **UC16 - Visualizza spese:** Voglio poter visualizzare le spesa di gruppo.
- **UC17 - Ricalcolo debiti:** Voglio poter calcolare i debiti.
- **UC18 - Inserisci spesa personale:** Voglio poter inserire una spesa personali.
- **UC19 - Elimina spesa personale:** Voglio poter eliminare una spesa personali.
- **UC20 - Modifica spesa personale:** Voglio poter modificare una spesa personali.
- **UC21 - Visualizza spesa personale:** Voglio poter visualizzare le spesa personali.
- **UC22 - Inserisci budget:** Voglio poter inserire un budget di risparmio.
- **UC23 - Elimina budget:** Voglio poter eliminare un budget di risparmio.
- **UC24 - Modifica budget:** Voglio poter modificare un budget di risparmio.
- **UC25 - Visualizza budget:** Voglio poter visualizzare i budget di risparmio.

### 1.3 Priorità casi d'uso

I casi d'uso possono essere suddivisi in tre categorie, a seconda della loro priorità nel processo di sviluppo:

- **Coda ad alta priorità:** Contiene i requisiti essenziali per il corretto funzionamento dell'applicativo. Questi includono la creazione dei profili utente, la creazione di nuovi gruppi spese, con le funzionalità annesse(invita,elimina membri ecc ecc) ed infine la gestione spesa con l'algoritmo di calcolo debiti.

- **Coda a media priorità:** Questa coda include funzionalità di supporto, principalmente orientate alla gestione spese personali.
- **Coda a bassa priorità:** Contiene le funzionalità meno rilevanti ossia la tematica del budget , per le quali non è prevista una loro implementazione immediata. Tuttavia, potrebbero essere implementate in futuro, a seconda dell'andamento del progetto.

ID	Titolo
UC1	Login
UC2	Registrazione
UC3	Logout
UC8	Invita memebri
UC9	Elimina membri
UC10	Modica memebri
UC11	Elimina gruppo
UC12	Accedi gruppo
UC13	Inserisci spesa
UC14	Elimina spesa
UC15	Modifica spese
UC16	Visualizza spese
UC17	Ricalcolo debiti

Tabella 1.1: Coda alta priorità

## 1.4 Architettura

L'architettura del sistema è basata su un'architettura a microservizi con i seguenti componenti principali:

- **Client/Frontend:** Implementato in React.js, il frontend fornisce un'interfaccia utente intuitiva per la gestione delle spese.

ID	Titolo
UC4	Crea alert di gruppo
UC5	Modica alert
UC6	Elimina alert
UC18	Inserisci spesa personale
UC19	Elimina spesa personale
UC20	Modica spesa personale
UC21	Visualizza spesa personale

Tabella 1.2: Coda media priorità

ID	Titolo
UC22	Inserisci budget
UC23	Elimina budget
UC24	Modica budget
UC25	Visualizza budget

Tabella 1.3: Coda media priorità

- **API Gateway:** Un punto di ingresso unico per tutte le richieste del client, responsabile dell'instradamento verso i microservizi corretti.
- **Backend (Microservizi):** I microservizi sono sviluppati in Spring Boot e includono:
  - **Servizio Utente:** Gestisce la registrazione, l'autenticazione e il profilo utente.
  - **Servizio Gestione Spese:** Gestisce le operazioni CRUD sulle spese.
  - **Servizio Gruppi:** Gestisce la creazione e la modifica dei gruppi.
- **Database:** Utilizzo di PostgreSQL per la gestione dei dati relazionali.



## 1.5 Toolchain

Per la realizzazione della piattaforma **Spendly** verranno utilizzati i seguenti strumenti:

### Modellazione

- Diagrammi (casi d'uso, deployment, componenti, classi, flussi, entità-relazione): **Draw.io**, **StarUML**.

### Implementazione Applicazione Client

- Linguaggio di programmazione: **Java**;
- IDE: **Visual Studio Code**;
- Framework: **Spring**.

### Implementazione Web Server

- Linguaggio di programmazione: **Java**;
- IDE: **Visual Studio Code**;
- Framework: **Spring**;
- Analisi statica: **Checkstyle**, **SonarLint**;
- Testing e analisi dinamica: **JUnit**.

### Implementazione Database

- Tipologia: **Relazionale**;
- Database: **MySQL**;
- Gestione: **MySQL Workbench**.

## Documentazione, Versioning e Organizzazione del Team

- Documentazione: **LaTeX**;
- Versioning: **GitHub**;
- Git Client: **GitHub Desktop**;
- Organizzazione del Team: **Microsoft Teams**, **Trello**.