
Software Requirements Specification

per

Progetto Academy

Version 1.0 Approved by Fabrizio Frate

Prepared by Marco Pecorella, Marco Nardi, Davide Giordano

Capgemini

Table of Contents

1. Introduzione	1
1.1. Scopo.....	1
1.2. Consigli e Suggerimenti per la produzione.....	1
1.3. Scadenza suggerite e Milestone.....	1
2. Descrizione	1
2.1. Contesto e Descrizione del prodotto.....	1
2.2. Classi e Descrizione.....	2
2.3. Ambiente di Sviluppo.....	2
2.4. Obblighi di Design.....	2
3. Features di Sistema	2
3.1. System Feature 1.....	2
3.2. System Feature 2.....	3
3.3. System Feature 3.....	3
3.4. System Feature 4.....	3
3.5. System Feature 5.....	3
4. External Interface Requirements	3
4.1. User Interfaces.....	3
4.2. Communications Interfaces.....	3

Revision History

Name	Date	Reason For Changes	Version
Marco Pecorella	11/06/22	Prima Produzione	0.1
Marco Nardi	11/06/22	Correzione	0.2
Davide Giordano	11/06/22	Correzione	1.0

1. Introduzione

1.1. Scopo

- L'applicativo proposto serve per gestire e monitorare il ciclo di vita dei thread ai quali viene assegnato un pacchetto di lavoro.
- L'esercizio proposto serve per migliorare le skills di utilizzo dei QThread e lavorare sulle skills di produzione GUI con QT.

1.2. Consigli e Suggerimenti per la produzione

Nel corso del documento saranno presentati tutti i requisiti, al piè di pagina saranno forniti suggerimenti e consigli sul come procedere sull'attività descritta.

Si consiglia prima di effettuare una lettura completa del documento, comprendere al meglio la richiesta e poi procedere nei prossimi step.

Per domande e consigli, saremo ben lieti di rispondervi tramite mail:

Giordano, Davide <davide.giordano@capgemini.com>

Nardi, Marco <marco.nardi@capgemini.com>

Pecorella, Marco <marco.pecorella@capgemini.com>

2. Descrizione

2.1. Contesto e Descrizione del prodotto

L'applicativo dovrà essere composto da una finestra nella quale l'utente avrà la possibilità di associare un determinato algoritmo (pacchetto di lavoro) ad una difficoltà computazionale.

Tramite un click del bottone "Start", l'applicativo assocerà, ad un thread, un pacchetto di lavoro con l'algoritmo e la difficoltà specificati. L'utente potrà vedere lo stato di avanzamento di quest'ultimo tramite una progress bar collegata al pacchetto di lavoro associato.

L'utente avrà la possibilità di vedere i log dei vari thread all'interno di un Widget.

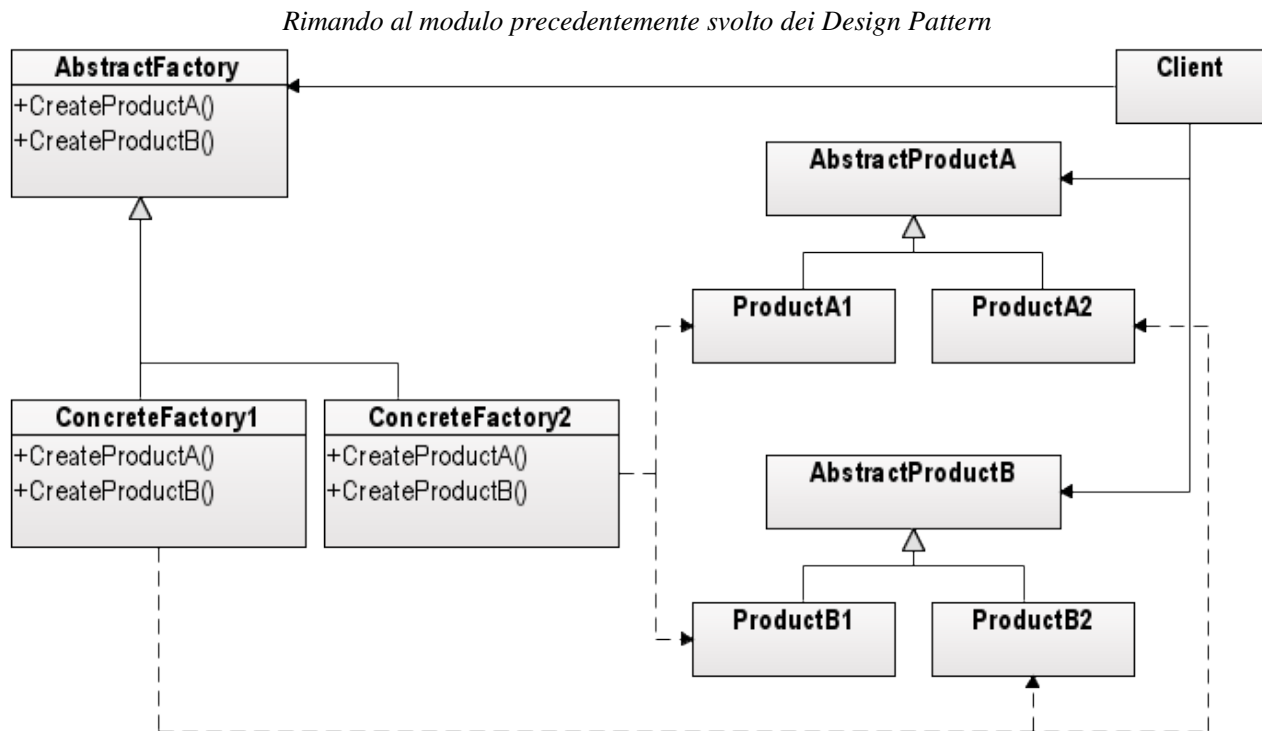
2.2. Ambiente di Sviluppo

L'ambiente di Sviluppo deve essere Qt e l'ambiente di Design deve essere QtCreator5.

2.3. Obblighi di Design

Per produrre i pacchetti di lavoro, deve essere utilizzata una Abstract Factory.

Di seguito è presente il diagramma UML del design pattern:



3. Features di Sistema

3.1. SRS #1 Scelta algoritmo

Il sistema deve permettere all'utente di selezionare un algoritmo*

3.2. SRS #2 Scelta difficoltà

Il sistema deve permettere all'utente di selezionare una tra le seguenti difficoltà di calcolo dell'algoritmo:
easy, medium, hard

3.3. SRS #3 Creazione Thread

Il sistema deve permettere all'operatore di associare un thread al pacchetto di lavoro selezionato.

3.4. SRS #4 Gestione progresso pacchetto di lavoro

Il sistema deve fornire in output all'operatore il progresso del pacchetto di lavoro **mediante progress bar univoca associata ad una label**.

3.5. SRS #5 Gestione dei Log

Il sistema deve fornire in output all'operatore lo stato corrente del thread: **created, working, completed**.

* *Puoi scegliere gli algoritmi che preferisci purché sia possibile avere un tracciamento del loro stato, suggeriamo: bubble sort, merge sort, ordinamento di una lista linkata*

NB: Il Log deve essere mostrato nel Widget dedicato precedentemente menzionato

4. External Interface Requirements

4.1. User Interfaces

L'applicativo deve essere in **dark theme**.*

Il font utilizzato deve essere **Arial**.

I bottoni devono essere **rounded**

4.2. Communications Interfaces

La comunicazione tra istanza di oggetti deve essere fatta mediante **SIGNAL & SLOT**

* *Si suggerisce l'uso di un file qss*