"Software Engineering"

Course

a.a. 2016-2017

Template version 1.0

Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)

Planner Path Calculator version v2 Deliverables

Index

Project Rules and Evaluation Criteria

Date	27/11/2016
Deliverable	*Deliverable 1*
Team (Name)	Indifferente

Team Members		
Name & Surname	**Matriculation Number**	**E-mail address**
Tommaso Di Salle	*236202*	*Tommasodisalle@gmail.com*
Luca D'Orazio	*227635*	*Lucaadorazio@gmail.com*
Stefano Corsetti	*227288*	*s.corsetti@hotmail.it*
Francesco Di Cato	*203356*	*dicatofrancesco@gmail.com*
Eugenio Mancini	*230024*	*Emancini1992@libero.it*

Table of Contents of this deliverable

Punti A , B , C , F , H.

2.List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Studio del progetto e metodi organizzativi	22/10/2016	24/10/2016	Abbiamo analizzato il testo e cercato di capire cosa il Sistema deve fare. Le comunicazioni avverranno attraverso skype e la condivisione attraverso drive e github
Utlizzo Software MagicDrow	8/11/2016	8/11/2016	Consultazione manuale online
Deduzione/estrapolazione requisiti funzionali	2/11/2016	3/11/2016	Siamo stati in grado di riconoscere subito le funzionalità/servizi del Sistema
Requirement elicitation	7/11/2016	7/11/2016	Attraverso il file condiviso abbiamo dedotto I requisiti funzionali
Studio e preparazione use-case diagram	10/11/2016	17/11/2016	Risolti i dubbi chiedendo al committente

Analysis model	18/11/2016	24/11/2016	Abbiamo analizzato I tre gruppi: BoundaryObject, Entity Object e ControllerObject. Richiesti chiarimenti al docente e approfondimento su libro di testo
Architettura SW	22/11/2016	26/11/2016	Approfondimenti su libro ti testo, slide ed internet su Component Diagram.
Document/ Documentazione	24/11/2016	26/11/2016	Il team ha documentato tutte le scelte fatte in linguaggio natural

A. Requirements Collection

A.1 Functional Requirements

Lista Requisiti funzionali

1- Creazione Albero
    1.1-Parametri
        1.1.1- Scelta split-size
        1.1.2- Scelta depth
        1.1.3- Lista attributi
            1.1.3.1- archi
            1.1.3.2- nodi
        1.1.4- Scelta nomi
            1.1.4.1- Nome albero
            1.1.4.2- Nome vertici
    1.2- Salva Albero
2- Seleziona Albero
    2.1- Carica Albero
    2.2- Cancellazione
3- Calcolo Somma
    3.1- Scelta due vertici
    3.2somma tra i 2 vertici
4- tempo di calcolo

Requisiti funzionali

I numeri (*) rispettano l'ordine della lista dei requisiti funzionali.

Dall analisi dei requisiti e dal file condiviso abbiamo estrapolato i seguenti R.F. che possono essere raggrupati in :
Creazione, caricamento, cancellazione, seleziona e calcolo.

Creazione(1) dedotto da "**The PPC must have a GUI that generates a tree structure using the following parameters**". Da questa frase vengono a cascata tutti i punti di (1.1.*) che sono tutte i parametriche che deve avere l'albero

SalvaAlbero(1.2) :L'albero verrà generato e salvato cliccando sul bottone "buildtree". Requisito dedotto da "The GUI can be used to generate a tree when the "Build Tree" button is pressed. The result is saved in the database."

Carica Albero(2.1): l''utente può scegliere un albero precedentemente salvato. Dedotto da "The GUI can be also used to retrieve from the database a Tree previously stored"

Cancellazione(2.2) dedotto dal file id domanda 6 con risposta "L'albero puo' essere inserito, cancellato e visualizzato. Non serve modificarlo."

Seleziona(2): dedotta dal file id domanda 6("L'albero puo' essere inserito, cancellato e visualizzato. Non serve modificarlo.") e da id domanda 22("Non e' necessario visualizzare l'albero in quanto stiamo parlando di alberi potenzialmente grandissimi. Se trovate un modo efficiente di farlo va benissimo, ma consideratela una cosa opzionale. Quello che e' richiesto invece e' visualizzare il risultato dell'operazione di calcolo con la lista dei

vertici attraversati (es: Vertice3, Vertice4, Vertice 5. Ptime=50 Cost=210)"). Abbiamo quindi deciso di inserire una funzionalità antecedente quella di caricamento e cancellazione in cui l'utente avrà una descrizione generale dell'albero(altezza, splitsize, numero totale nodi etc,) e da questa potrà scegliere se cancellare o modificare l'albero.

CalcoloSomma(3): che raggruppa Scelta di 2 vertici (3.1) e somma tra i 2 vertici(3.2) dedotto da "users can select **2 Vertices A and B, and the system must return the list of vertices from A to B among with the SUM of each attribute**

Tempo di calcolo(4) dedotto da "The GUI will also display the time the system took to make this calculation" **.

** per tempo di calcolo non intendiamo che il Sistema deve essere veloce a ritornare il risultato(NFR) ma che come funzionalità deve riportare a video il tempo impiegato

A.2 Non Functional Requirements

-Il Sistema deve essere veloce a ritornare i risultati;
-Gestire 10/100 utenti senza subire cali di prestazioni;
-Gestire alberi fino a 2000000 di nodi;
-Gui in html5;
-La sicurezza non deve essere garantita;

A.3 Content

1) I dati verrano immessi dall' utente per creare un nuovo albero.
2) Per recuperare un albero, i dati verrano estratti da un database.

A.4 Assumptions

1) Assumiamo che il Sistema sia collegato in rete.
2) Assumiamo che il Sistema disponga di risorse di calcolo adeguate.
3) Si assume che il calcolo venga effettuato tra due nodi di cui uno deve essere antenato del secondo

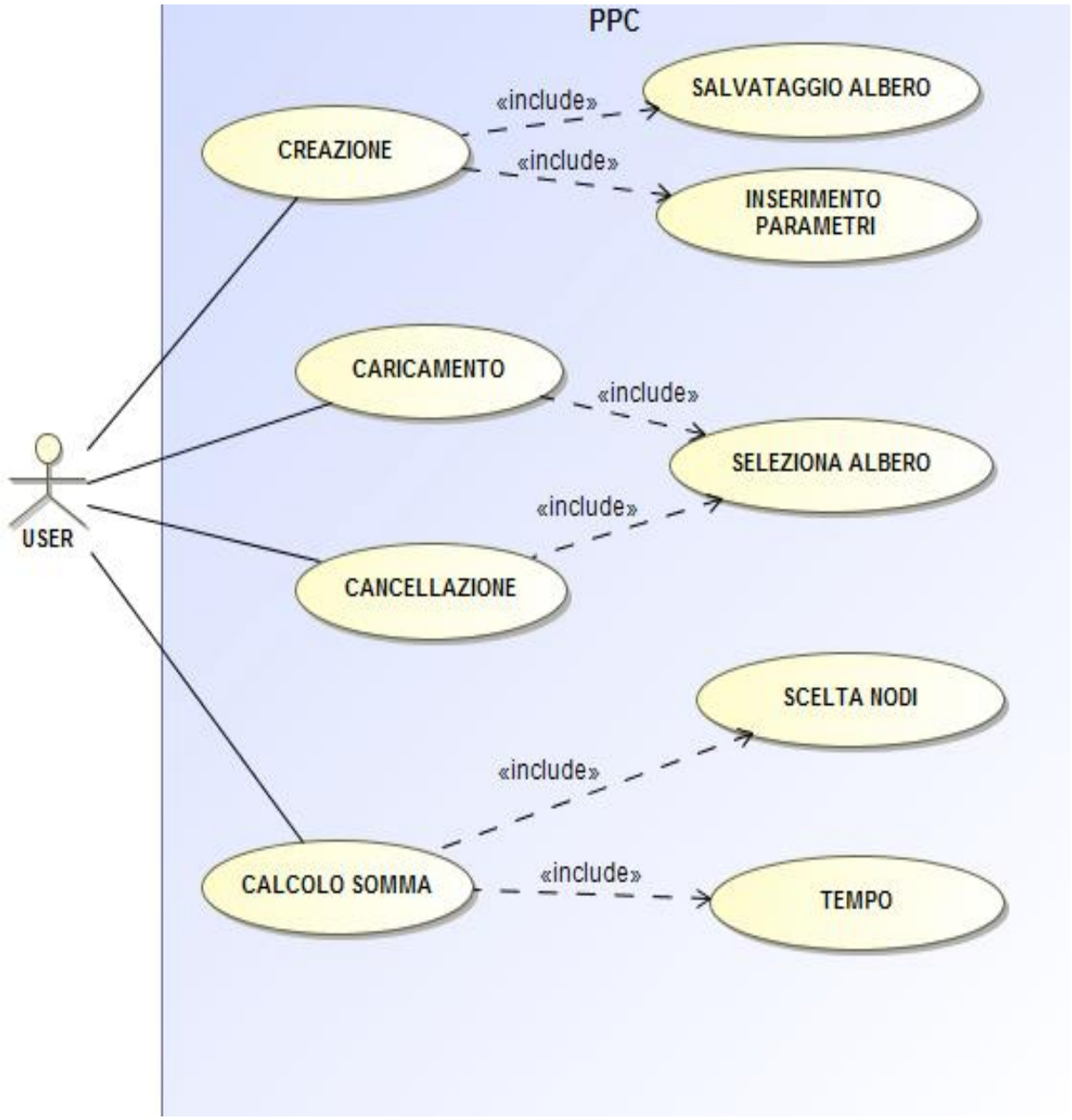
A.5 Prioritization

Requisiti listati secondo ordine di priorità.

Requirement	**Priorità**	**Motivazione**
CalcoloSomma	High	Funzione indispensabile del sistema.
Creazione Albero	High	Senza questa funzionalità, si creerebbero alberi con poca utilità per l'utente.
InserimentoParametri	Medium	Funzionalità indispensabile richiesta dal cliente.
Caricamento	Medium	È necessario per eseguire altre funzionalità. Permette di caricare alberi già presenti nel sistema,
Cancellazione	Low	È una funzionalità opzionale, poiché senza di essa è comunque possibile effettuare le altre funzionalità (operazioni).

A.6 *USE-CASE Diagram*

Abbiamo deciso di fermaci ad un alto livello di astrazione. Abbiamo identificato solo i 4 use-case principali che sono emersi dai vari scenari ipotizzati e alcuni di questi sono stati esplosi con le funzionalità che riteniamo più importanti.



A.7 *Scenari :*

Di seguito alcuni possibili scenari:

1- CREA:

l'utente inserisce i parametri dell'albero (nomi, split-size e altezza).

Sceglie la lista di attributi da inserire a nodi ed archi ed il range di valori da assegnare ad ogni attributo. Cliccando sul pulsante BUILD-TREE verrà generato l'albero con tali proprietà e salvato.

2- SELEZIONA:

l'utente seleziona l'albero da un elenco presente nella lista e avrà informazioni sull' albero

3- CALCOLO SOMMA:

l'utente sceglierà 2 vertici. Tra questi 2 vertici verrà calcolata la somma di tutti gli attributi su archi e nodi e il risultato verrà stampato a video insieme al tempo impiegato per effettuare l'operazione

4- CANCELLAZIONE

L'utente cliccando sul bottone "cancella" elimina l'albero selezionato dall'elenco

A.8 *UseCase Description*

USE CASE	Creazione	
Goal in Context	Generazione e Salvataggio Albero	
Scope & Level	L'utente sceglie i valori da assegnare all'albero e quest'ultimo viene salvato	
Preconditions	Inserimento Parametri corretti	
Success End Condition	L'albero viene generato	
Failed End Condition	(Notifica di errore)	
Primary Actor	USER	
Trigger	Click sul pulsante "CREA"	
DESCRIPTION	**STEP**	**ACTION**
	1	L'utente inserisce i parametri (split-size, altezza, nome, lista attributi nodi/archi e range valori per ogni attributo)
	2	Salvataggio albero generato

RELATED INFORMATION	Creazione
Priority	High
Performance	< 2 ore
Frequency	1 volta a settimana per utente
OPEN ISSUES	• Metodologia salvataggio albero • Metodologia rappresentazione albero

USE CASE	Seleziona Albero	
Goal in Context	Selezione di un albero tra quelli presenti	
Scope & Level	L'utente cliccando su un albero tra la lista di quelli presenti recupererà le informazioni ad esso associate (numero nodi, altezza, lista attribute, etc..)	
Preconditions	L'utente ha cliccato su "Lista"	
Success End Condition	è stato selezionato un albero	
Failed End Condition	(notifica di errore)	
Primary Actor	USER	
Trigger	Click su albero	
DESCRIPTION	**STEP**	**ACTION**
	1	Click per la selezione di un albero

RELATED INFORMATION	Seleziona albero
Priority	Low
Performance	< 5 sec
Frequency	(caricamento) 3 volte al giorno per utente (cancellazione) 1 volta a settimana per utente
Superordinates	• Caricamento • Cancellazione

USE CASE	Calcolo Somma	
Goal in Context	Somma sugli attributi dei nodi selezionati	
Scope & Level	il Sistema fornirà la lista dei vertici tra i nodi selezionati e la somma dei valori degli attributi su nodi/archi di tale lista	
Preconditions	L'utente ha cliccato sul bottone "carica"	
Success End Condition	Stampa a video il risultato della somma e il tempo impiegato per calcolarlo	
Failed End Condition	Notifica Errore	
Primary Actor	USER	
Trigger	Click sul pulsante "SOMMA"	
DESCRIPTION	**STEP**	**ACTION**
	1	L'utente sceglie due nodi
	2	Clicca sul bottone calcola
	3	Il sistema ritorna a video la lista dei vertici, il risultato della somma ed il tempo impiegato a calcolarla

RELATED INFORMATION	Calcolo somma
Priority	High
Performance	< 30 sec per 1 milione di nodi <60 sec per 2 milioni di nodi
Frequency	3 volte al giorno per utente
OPEN ISSUES	Le operazioni verranno fatte sul DB o sul server
Subordinates	• Scelta nodi • Tempo

B. Analysis Model

B.1 *Robustness Diagram*

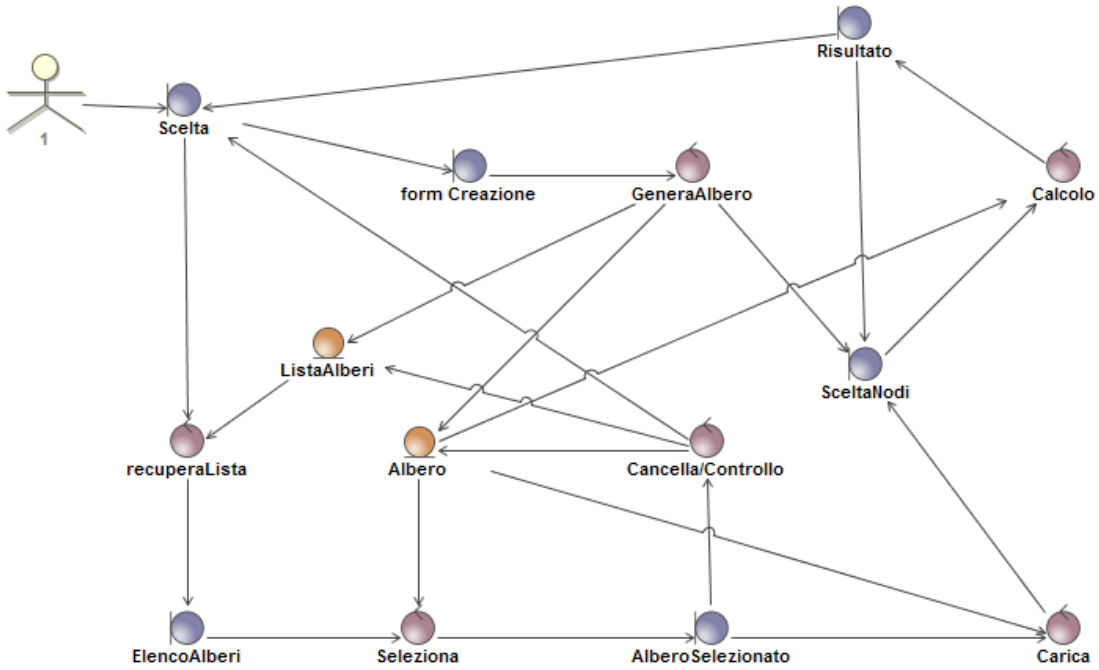
Il team ha prodotto il diagramma di Analysis Model affrontando diverse decisioni.

Come boundary abbiamo identificato i seguenti oggetti:
Scelta, form Creazione, Elenco Alberi, Albero selezionato, Scelta nodi, Risultato.

Come controller, abbiamo identificato i seguenti oggetti: Recupera lista, Seleziona, Carica, Cancella/Controllo, Calcolo, Genera albero.

Come entity abbiamo identificato i seguenti oggetti: Lista alberi ed Albero.

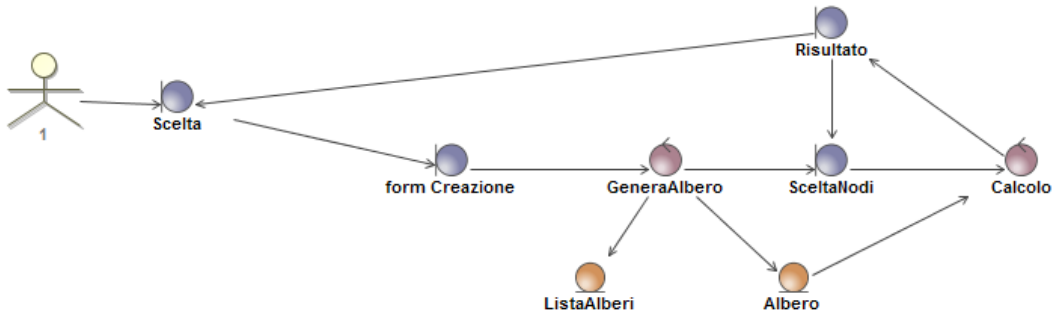
Come primo step l'utente si interfaccia con una schermata, identificata dal boundary "Scelta", che rappresenta la pagina iniziale della nostra applicazione. L'utente avrà due opzioni: creare un nuovo albero cliccando sul bottone "Crea", oppure visualizzare la lista degli alberi presenti nel Database, cliccando sul bottone "Lista".



Ramifichiamo la spiegazione in base al bottone premuto e riportando il relativo diagramma.

Caso1

Bottone Crea:

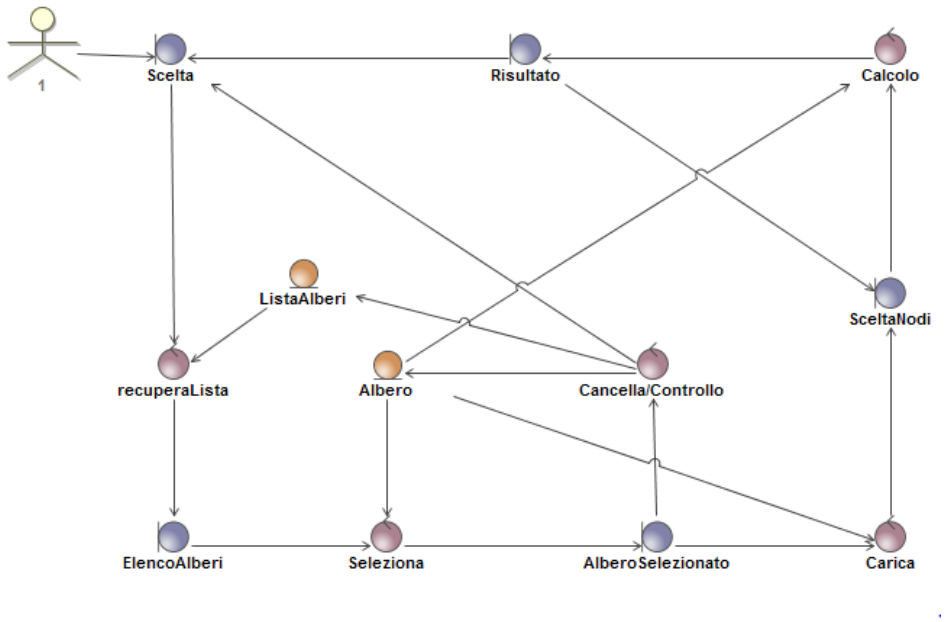


-Form creazione: Boundary Object dove l'utente inserisce I parametri per la generazione dell'albero.

-GeneraAlbero: Controller che prende I parametri inseriti in FormCreazione e crea l'albero

Caso2

Bottone Lista:



-RecuperaLista: l'utente, precedentemente ha cliccato su "Lista", questo Controller ha la funzionalità di recuperare le informazioni presenti in "ListaAlberi"

-ElencoAlberi: Entity che mostra l'elenco degli alberi presenti in "ListaAlberi" .Cliccando su un albero presente nella lista entra in funzione il controller "Seleziona"

-Seleziona: Controller che recupera informazioni sull'albero

-Albero Selezionato: Entity che mostra le informazioni dell'albero precedentemente selezionato. Avrà due bottoni, di cui uno "Carica" e uno "Cancella".

-Carica: Controller precedentemente citato, che ha la funzione di caricare l'albero

-Cancella: Controller che ha la funzione di eliminare definitivamente l'albero precedentemente selezionato

Object in commune tra i 2 percorsi:

-ListaAlberi: Entity Object in cui è inserita la lista dei nomi degli alberi presenti.

-Albero: Entity Object in cui è inserita la lista dei nodi ed il valore degli attributi

---

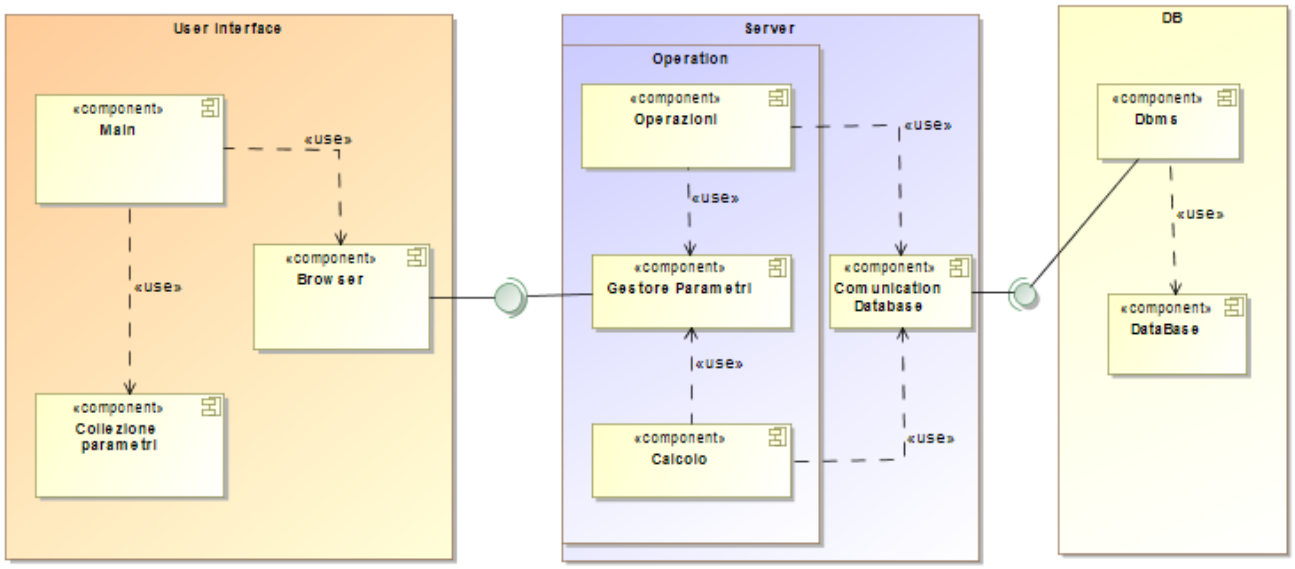
-SceltaNodi: Entity Object dove l'utente sceglie I due vertici e gli attributi su cui effettuare la somma

-Calcolo: Controller Object che ha la funzione di effettuare il calcolo della somma e l'elenco dei nodi tra I due selezionati

-Risultato: Boundary Object che mostra a video il risultato calcolato dal controller "Calcolo" ed il tempo impiegato per effettuare questa operazione.
Qui ci sono due bottoni "Pagina iniziale" e "Ricalcola".
Cliccando su "PaginaIniziale" l'utente viene portato nell'Entity "Scelta".
Cliccando su "Ricalcola" l'utente viene portato nell'Entity "SceltaNodi" ed effettuare nuovamente calcolci sull'albero precedentemente selezionato

C. Software Architecture

C.1 *The static view of the system: Component Diagram*



C1.1 *Documentazione component diagram*

Nel design del sistema abbiamo utilizzato un architettura client/server. La componente UserInterface(U.I.) richiede dei servizi alla componente Server che risponderà alle richieste interrogando la componente DB. Quindi ci sono 3 macro: U.I., Server e DB.

1) User Interface:

la component U.I. è l'interfaccia con cui comunica l'utente.
A sua volta è stata divisa in 3 component:
La prima, chiamata browser, si occuperà di creare/gestire le finestre e da interfaccia con il Server.
A sua volta il Browser gestisce 2 sottosistemi che sono Main e CollezioniParametri le quali rispettivamente si occupano di far visualizzare le pagine in cui l'utente non può inserire/ricevere dati e le pagine in cui l'utente inserisce/riceve dati dalla componente Server.

2)Il Server è la componente che si occupa di servire le richieste della U.I. E' la parte principale del nostro sistema. E' stata divisa in 2 sottosistemi:

2.1) Operation,

a sua volta diviso in:

- Gestore Parametri: sottosistema che si occuperà di smistare le richieste;
- Operazioni:sottositema che si occupera di generare,salvare,cancellare, recuperare Lista e informazioni sugli Alberi;
-Calcolo: sottosistema che si occupera di effettuare l'operazione di somma;

2.2) ComunicationDatabase: è il sottosistema che interroga il DB. Esso deve gestire eventuali race-condition nella creazione e cancellazione di un albero.

3) L' ultima componente DB sarà composta da un sottosistema dbms che si occuperà di gestire e recuperare i dati e una parte Database che sarà quella dove saranno realmente memorizzati.

F. Design Decisions

1) L'operazione di creazione impiega grande quantità di tempo, non tanto in termini di computazione, quanto in termini di scrittura su disco.
Da specifiche “il risultato deve essere salvato su database”.
Il Cliente non è interessato ai tempi di caricamento e come ma abbiamo constatato che anche un albero di medie misure, risulta essere molto pesante in termini di dimensione. Ciò potrebbe infastidire la user-experience.
Quindi cercheremo di ridurre al minimo i tempi di creazione.

2) L’operazione di calcolo/somma deve essere il più veloce possibile.
Stiamo cercando di sfruttare la completezza dell’albero (ogni nodo ha lo stesso numero di figli), utilizzando la relazione spiegata nell’esempio che segue

Per split-size= 2

calcolo figli figlio SX Numeronodo*split-size ,
figlio DX (NumeroNodo*split-size)+1

Questa soluzione è molto efficace in quanto al + richiede l’altezza massima dell’albero di interrogazioni ma considerando che le specifiche potrebbero cambiare, e all’aumentare della split-size aumenta anche il numero dei controlli da effettuare, abbiamo deciso di non utilizzarla o al massimo di tenerla come alternativa e di adottare altre soluzioni, trovate con l’aggiunta di informazioni sull’albero.

Possibile soluzione 1:
Per ogni elemento(nodo) aggiungeremo informazioni sul padre
Possibile soluzione2:
per ogni elemento(nodo) memorizzeremo tutto il “path” fino alla radice.

3) Il team ha deciso di adottare la componente "Comunication Database" per far comunicare tutte le operazioni con il Database.
Questa componente è necessaria per la comunicazione con il DB, ma soprattutto per la gestione della concorrenza.
In lettura non vi è nessun tipo di problema ma in fase di creazione e di cancellazione dobbiamo garantire che non vengano a verificarsi race-condition.
Possibili soluzioni: utilizzo dei semafori.

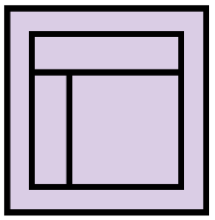
4) Utilizzare la funzione Seleziona per evitare due interfacce grafiche identiche con la sola differenza dell’operazione da effettuare (cancellazione o caricamento).
Quando si andrà a selezionare un albero tra quelli presenti nella lista fornita (quindi già presente nel Database) l'utente avrà la possibilità di vedere a video le caratteristiche di tale albero.

5) Poichè il sistema sarà di tipo client-server utilizzeremo un la classica configurazione AMP. Se non adottabile utilizzeremo il linguaggio di programmazione Java. NOTA: abbiamo creato il component diagram in modo che anche se impossibilitati ad utilizzare un webserver (apache) lo schema si adattasse ad altre soluzioni. Le modifiche riguarderebbero solo la componente GestoreParametri (lato Server).

6) Il team ha deciso che il calcolo della somma verrà gestito dal server.
A meno di drastici cambiamenti dei requisiti la soluzione di effettuare la somma sul server e non sul DB ci sembra la più sensata.

7) Abbiamo deciso di non inserire un gestore degli errori adottando tale soluzione: eventuali errori di inserimento parametri verranno gestiti dal componente CollezioneParametri, eventuali errori in fase di creazione dell albero dalla componente Operazioni ed eventuali errori di salvataggio dalle componenti ComunicationDataBase/Operazioni.

H. Effort Recording



GANTT

Personal Journal

Team: INDIFFERENTE

Student Name: Tommaso di Salle, Luca Alessandro D'orazio, Stefano Corsetti, Francesco di Cato, Eugenio Mancini

Student number:5

email: tommasodisale@gmail.com, lucaadorazio@gmail.com, s.corsetti@hotmail.it, dicatofrancesco@gmail.com, emancini1992@libero.it

When	Time Spent	Partners	Brief description of the performed task	Category	Sub-Category
22/10/16	3 ore	5	Studio del progetto e metodo organizzativo Nel primo incontro il gruppo ha analizzato il testo per comprendere le richieste del progetto. Il gruppo comunicherà tramite: Skype, Telegram. Per rimanere aggiornati ogni modifica Verrà pubblicata su google drive.		
02/11/16	2 ore	3	Utilizzo software MagicDraw	Doing	
07/11/16	2 ore	5	Deduzione/estrapolazione requisiti funzionali	Learning	
08/11/16	2 ore	5	Requirement elicitation	Doing	
10/11/16	5 ore	4	Studio e preparazione use-case diagram	Doing	
18/11/16	4 ore	5	Analysis model	Doing	
22/11/16	5 ore	5	Architettura SW	Doing	
24/11/16	2 ore	3	Document/ Documentazione	Doing	
25/11/16	2 ore	5	Divisione del lavoro	Learning	