

**O**bject

**D**esign

**D**ocument

|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 0.1 |
| Data | 19/12/2020 |
| Destinatario | Prof.ssa F. Ferrucci |
| Presentato da | **H**ermann Senatore, **I**van Carmine Adamo, **L**orenzo Criscuolo, **O**razio Cesarano |
| Approvato da |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | Versione | Cambiamenti | Autori |
| 19/12/2020 | 0.1 | Prima Stesura | [tutti] |

Sommario

[1. Introduzione 2](#_Toc59286893)

[1.1. Object Design Trade Off 2](#_Toc59286894)

[1.2. Components Off the Shelf (COTS) 2](#_Toc59286895)

[1.3. Interfaces, Documentation, Guidelines 3](#_Toc59286896)

[1.3.1. Classi e Interfacce Java 3](#_Toc59286897)

[1.3.2. Pagine Lato Server 3](#_Toc59286898)

[1.3.3. Pagine HTML 3](#_Toc59286899)

[1.3.4. Script JavaScript 3](#_Toc59286900)

[1.3.5. Fogli di Stile CSS 3](#_Toc59286901)

[1.3.6. DB SQL 3](#_Toc59286902)

[1.4. Design Pattern e Architectural Pattern 3](#_Toc59286903)

[1.4.1. MVC (Model View Control) 3](#_Toc59286904)

[1.4.2. Da decidere 1 3](#_Toc59286905)

[1.4.3. Da decider 2 3](#_Toc59286906)

[1.4.4. Da decider 3 3](#_Toc59286907)

[1.5. Definizioni, Acronimi e Abbreviazioni 3](#_Toc59286908)

[2. Packages 3](#_Toc59286909)

[2.1. Interface 3](#_Toc59286910)

[2.2. Model 3](#_Toc59286911)

[2.3. View 3](#_Toc59286912)

[2.4. Control 3](#_Toc59286913)

[3. Class Interfaces 3](#_Toc59286914)

[4. Class Diagram 3](#_Toc59286915)

[5. Glossario 3](#_Toc59286916)

# Introduzione

## Object Design Trade Off

### Usabilità vs Funzionalità

Il Sistema dovrà prediligere l’usabilità a discapito delle funzionalità previste nella fase di Analisi in quanto risulta più prioritario fornire un sistema user friendly a discapito di operazioni superficiali.

### Costo vs Robustezza

Il Sistema sarà sviluppato in modo robusto a discapito dei costi in quanto essendo prettamente di uso medico può essere considerato “Mission Critical”. Per questo motivo si preferisce sostenere costi maggiori al fine di ottenere un sistema robusto.

### Efficienza vs Portabilità

Il Sistema dovrà favorire una maggiore efficienza a discapito della portabilità. Questa scelta nasce dall’esigenza di avere un sistema snello e in grado di eseguire operazioni nel miglior modo e nel minor tempo possibile al fine di suscitare fiducia negli utenti finali del sistema.

### Sviluppo Rapido vs Funzionalità

Il Sistema sarà sviluppato con un minor di funzionalità per favorire uno sviluppo rapido, in quanto sono presenti delle deadline e non è presente abbastanza tempo per implementare anche tutte le funzionalità ritenute meno importanti.

### Costo vs Riusabilità

Il Sistema proposto, essendo realizzato ex novo, non avrebbe senso parlare di riutilizzo di componenti già esistenti. Per tanto non si può ignorare la necessità di sostenere costi maggiori per lo sviluppo

### Tempo di Risposta vs Affidabilità

Il Sistema dovrà garantire una maggiore affidabilità a discapito del tempo di risposta su operazioni critiche in quanto deve essere ridotta al minimo la possibilità di introdurre errori o incongruenze nei dati dovute alla gestione della concorrenza, ecc.

## Components Off the Shelf (COTS)

Il Sistema utilizzerà I segunti componenti off the shelf:

* Bootstrap, un framework per aiutare lo sviluppo delle interfacce grafiche che utilizza HTML, CSS e JS;
* Vue.js, un framework per JS per semplificare la validazione dell’input e in generale del front end;
* Spring, un framework scritto in Java che semplifica la gestione lato back end e in generale aiuta nello sviluppo di applicazioni basate su MVC;
* JUnit, un framework per agevolare lo unit testing per Java;
* Katalon, un plugin per browser per agevolare system testing;
* Apache Tomcat, un web server con annesso application container per applicazioni scritte in Java.

## Interfaces, Documentation, Guidelines

### Classi e Interfacce Java

### Pagine Lato Server

### Pagine HTML

### Script JavaScript

### Fogli di Stile CSS

### DB SQL

## Design Pattern e Architectural Pattern

### MVC (Model View Control)

### Da decidere 1

### Da decider 2

### Da decider 3

## Definizioni, Acronimi e Abbreviazioni

# Packages

## Interface

## Model

## View

## Control

# Class Interfaces

# Class Diagram

# Glossario