**.**

**UNIVERSITÀ DEGLI STUDI DI SALERNO**

**ANNO ACCADEMICO 2019/2020**

**REQUIREMENTS ANALYSIS DOCUMENT**

**Version 2.0**

**TOP MANAGER**

**Prof. Andrea De Lucia**

**PROJECT MANAGER**

**Giuseppe De Michele**

**Partecipanti**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Mario Balbi | 0512102944 |
| Giuseppe De Michele | 0512102642 |
| Singh Karanbir | 0512104924 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 12/11/2019 | 1.0 | Introduzione:  Scopo del sistema;  Ambito del sistema;  Obiettivi e criteri di successo del progetto; Panoramica; | De Michele Giuseppe |
| 14/11/2019 | 1.1 | Sistema corrente  Sistema proposto:  Panoramica;  Identificazione attori; | De Michele Giuseppe |
| 19/11/2019 | 1.2 | Requisiti non funzionali | De Michele Giuseppe  Balbi Mario |
| 19/11/2019 | 1.3 | Requisiti funzionali | Singh Karanbir |
| 19/11/2019 | 1.4 | Scenari GA | Balbi Mario |
| 26/11/2019 | 1.5 | Scenari e revisione | Singh Karanbir |
| 10/12/2019 | 1.6 | Tabelle casi d’uso | De Michele Giuseppe |

**Indice**

1. Introduzione
   1. Descrizione del Problema
   2. Design Goals
      1. DG\_0 Criteri di Performance
      2. DG\_1 Criteri di Affidabilità
      3. DG\_2 Criteri di Costo
      4. DG\_2 Criteri di Mantenimento
      5. DG\_2 Criteri di Utente
   3. Definizioni, Acronimi e Abbreviazioni
   4. Riferimenti
   5. Panoramica

**1. Introduzione**

* 1. **Object Design Trade-off**

Dopo aver stilato il documento di Requirements Analysis e il documento di System Design in cui vi è una descrizione sommaria di ciò che sarà il nostro sistema, definendo i nostri obiettivi ma tralasciando gli aspetti implementativi, andiamo ora a stilare il documento di Object Design che ha come obiettivo quello di produrre un modello che sia in grado di integrare in modo coerente e preciso tutte le funzionalità individuate nelle fasi precedenti.

In particolar modo, in tale documento si definiscono le interfacce delle classi, le operazioni, i tipi, gli argomenti e il signature dei sottosistemi definiti nel System Design. Inoltre, sono specificati i trade-off e le linee guida.

**Tempo di rilascio VS Manutenibilità:**

Il codice del sistema deve essere comprensibile il più possibile, in modo da facilitare la fase di testing ed eventuali future modifiche da apportare. Per rispettare queste linee guida il codice sarà accompagnato da commenti volti a semplificarne la comprensione. Ovviamente questo comporterà un aumento del tempo di sviluppo del nostro progetto

**Costi (ore uomo-lavoro) VS Robustezza:**

Non è stata garantita la robustezza del sistema per ridurre le ore uomo-lavoro. Non sono stati effettuati test dei guasti e non è stata verificata la predisposizione per andare su cloud. Ci siamo limitati ad effettuare test di carico con 500 utenti.

**Portabilità VS Efficienza:**

Abbiamo garantito la portabilità, in quanto richiesto dal cliente, utilizzando un linguaggio non nativo.

Java è indipendente dalla piattaforma, purtroppo questa caratteristica comporta prestazioni inferiori in quanto non viene compilato in linguaggio macchina.

**Throughput VS Sicurezza:**

La sicurezza, come descritto nei requisiti non funzionali del Requirements Analysis,

rappresenta uno degli aspetti importanti del sistema. Tuttavia, dati i tempi di sviluppo molto limitati, ci limiteremo ad implementare sistemi di sicurezza basati su username e password degli utenti.

**1.2 Linee guida per la documentazione dell’interfaccia**

**· Per il linguaggio Java usiamo la convenzione di Oracle: ·** https://www.oracle.com/technetwork/java/javase/overview/codeconvtoc-136057.html

**· Per le Servlet/JSP usiamo la convenzione di Oracle: ·** https://www.oracle.com/technetwork/articles/java/servlets-jsp-140445.html

**· Per il linguaggio SQL usiamo la seguente convenzione: ·** https://www.xaprb.com/blog/2008/10/26/the-power-of-a-good-sql-naming-convention/

**· Per i linguaggi HTML/CSS usiamo la convenzione di Google: ·** https://google.github.io/styleguide/htmlcssguide.html

**·Per il linguaggio JavaScript usiamo la convenzione di Google: ·** https://google.github.io/styleguide/jsguide.html