|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

***SocialBook  
Documento Analisi Requisiti***

****

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Barbato Alessia | 0512105858 |
| Proietto Angelica | 0512105762 |
| Russo Luca | 0512105840 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 26/11/2020 | 1.0 | Stesura introduzione (Paragrafi 1.1, 1.2, 1.3, 1.5, 1.6) | Russo Luca |
| 28/11/2020 | 2.0 | Stesura sistema corrente e proposto, requisiti non funzionali (Paragrafi 2, 3.1, 3.3.) | Russo Luca |
| 1/12/2020 | 3.0 | Stesura acronimi, requisiti funzionali e scenari (Paragrafi 1.4, 3.2, 3.4.1) | Proietto Angelica |
| 3/12/2020 | 4.0 | Stesura casi d’uso, realizzazione diagramma dei casi d’uso, tabella degli oggetti, sequence diagram (accesso, acquisto e relative eccezioni) | Barbato Alessia |
| 7/12/2020 | 5.0 | Realizzazione class diagram, sequence diagram (ricerca, registrazione, creazione ed eliminazione booklist, segui ed eliminazione utente e relative eccezioni) | Proietto Angelica |
| 11/12/2020 | 6.0 | Realizzazione sequence diagram (aggiunta al carrello), statechart diagram, mock-ups e navigational path | Russo Luca |
| 13/12/2020 | 7.0 | Glossario | Proietto Angelica |
| 18/12/2020 | 8.0 | Realizzazione activity diagram (sistema proposto) | Proietto Angelica |

Indice

1.Introduzione 4

1.1 Object Design trade-offs 4

1.2 Linee guida per la documentazione delle interfacce 8

1.3 Definizioni, acronomi, abbreviazioni 10

1.4 Design Pattern 11

1.5 Riferimenti 11

2. Packages

3. Interfacce delle classi 12

4. Glossario

1. Introduzione

Dopo aver stilato i documenti Requirements Analysis e System Design è necessario porre attenzione sugli aspetti implementativi. Questo documento ha l’obiettivo di produrre un modello che integri in modo coerente tutte le informazioni collezionate nelle fasi precedenti. In particolar modo, verranno definite le interfacce delle classi, le operazioni supportate, i tipi dei dati, i parametri delle procedure, le signatures dei sottosistemi definiti nel documento di System Design, i trade-offs e le linee guida.

* 1. Object Design trade-offs

***FUNZIONALITA’ vs AFFIDABILITA’***

L’integrazione di molteplici funzionalità potrebbe portare con maggiore facilità a malfunzionamenti di componenti software e quindi comprometterne drasticamente l’affidabilità.

***COMPRENSIBILITA’ vs TEMPO DI SVILUPPO***

Il codice del sistema deve essere comprensibile, in modo da facilitare la fase di testing ed eventuali future modifiche da apportare. Al fine di rispettare queste linee guida il codice sarà integrato da commenti volti a migliorarne la leggibilità; tuttavia questo richiederà una maggiore quantità di tempo necessario per lo sviluppo del nostro progetto.

**TEMPO DI RISPOSTA vs AFFIDABILITA’**

Per garantire che il sistema non si blocchi in caso di picchi di carico elevati, si preferisce rinunciare a un ottimo tempo di risposta. Si è deciso di tollerare questa problematica al fine di offrire agli utenti un sistema per l’appunto affidabile.

* 1. Linee guida per la documentazione delle interfacce

In questa sezione vengono definite le linee guida che ogni sviluppatore rispetterà per la leggibilità del codice. Per la formattazione dei file XML, HTML, CSS e JS si userà il formatter di IntelliJ di default, mentre per i file Java si seguiranno le convenzioni della Sun di Java . Tra cui evidenziamo alcuni aspetti fondamentali:

1.2.1 Naming Convention