|  |
| --- |
| Progetto di Ingegneria del Software 2015/16  U_F_L.jpg  **U.F.L.**  University Fanta League   * **Requirements Analysis Document -** |
| http://www.fisica.unisa.it/CentroQualita/images/logoUnisa.gif |

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Prof. De Lucia Andrea |  |

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Foresta Stefano | 0512100504 |
| Franzese Gennaro | 0512100270 |
| Paglialonga Giuseppe | 0510200478 |

|  |  |
| --- | --- |
| **Scritto da:** | Stefano Foresta, Giuseppe Paglialonga, Gennaro Franzese |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*Indice*

*Introduzione .1*

*Problema .2*

*Proposta .3*

*Scenari .4*

*Requisiti funzionali .5*

*Requisiti non funzionali .6*

*Ambiente di destinazione .7*

*Ambiente di sviluppo .8*

1. **Introduzione**
   1. **Descrizione del sistema**

Web application per la gestione di una fanta-lega semplice ed intuitiva. L’applicazione intende fornire i mezzi per realizzare un campionato automatizzato in grado di svolgere automaticamente i compiti che solitamente vanno svolti dal gestore di lega utilizzando mezzi quali carta e penna.

* 1. **Scopo del sistema**

Qualche settimana prima dell’inizio del campionato, per iniziare il fantacalcio, solitamente ci si incontra per fare l’asta e per redigere le informazioni sul campionato e le regole di lega. In ambito universitario può accadere, soprattutto tra studenti fuori sede, che è difficile incontrarsi e quindi difficile gestire l’inizio del gioco. Questa web application eliminirà il problema della distanza, che rallenta la costituzione di una lega, fornendo all’utilizzatore uno strumento che gli consente di partecipare anche a distanza e al gestore di lega di non dover perdere tempo facendo chiamate per organizzare l’incontro che serve a costituire le squadre e quindi la stessa lega.

* 1. **Obiettivi**

L’obiettivo e quello di fornire una piattaforma semplice ed intuitiva per la gestione di una semplice lega.

Le macro funzionalità che si vogliono offrire sono principalmente:

1. Registrazione utente e gestione del proprio profilo.
2. Creazione squadra e relativa gestione.
3. Gestione e visualizzazione delle varie leghe da parte del manager di lega.
4. Ricerca di un altro giocatore, visualizzazione profilo e invio/ricezione di un messaggio.
   1. **Definizioni, acronimi e abbreviazioni**

Vengono di seguito esplicitate definizioni, acronimi e abbreviazioni utilizzate nel documento per una lettura migliore.

|  |  |
| --- | --- |
| **Sigla** | **Significato** |
| PM | Project Manager |
| TM | Team Member |
| TL | Team Leader |
| PS | Problem Statement |
| ODD | Object Design Document |
| RAD | Requirements Analysis Document |
| SDD | System Design Document |
| TP | Test Plan |

* 1. **Overview**

Il presente documento RAD descrive il sistema in termini di requisiti funzionali e non funzionali e serve come base contrattuale tra cliente e consumatore.

1. **Architettura del sistema corrente**

L’architettura attualmente proposta non va a sostituire nessuna architettura esistente, il sistema è alla sua prima versione.

1. **Sistema proposto**
   1. **Requisiti funzionali**

In questa sezione vengono specificati quali funzioni il sistema deve fornire per soddisfare i bisogni dei gestori di lega e dei giocatori.

* + 1. **Requisiti funzionali per Amministratore**

|  |  |
| --- | --- |
| **Codice** | **Descrizione** |
| RF1 | Gestione leghe |
| RF2 | Gestione utenti |
| RF3 | Gestione voti |

**RF1 – Gestione leghe**

L’amministratore gestisce le leghe ossia le crea e apporta modifiche.

**RF2 – Gestione utenti**

L’amministratore accetta gli utenti che vogliono iscriversi ad una lega.

**RF3 – Upload voti**

L’amministratore settimanalmente inserisce nel database i voti relativi ai giocatori.

* + 1. **Requisiti funzionali per utente**

|  |  |
| --- | --- |
| **Codice** | **Descrizione** |
| RF4 | Gestione proprio account |
| RF5 | Gestione squadra |
| RF6 | Visualizzazione informazioni lega |

**RF4 – Gestione proprio account**

Creazione, modifica e gestione dell’account utente.

**RF5 – Gestione squadra**

Acquisto/Vendita giocatori, composizione squadra settimanale, scelta del modulo.

**RF6 – Visualizzazione informazioni lega**

Visualizzazione classifica, profili degli altri utenti e delle squadre da loro gestite.

* 1. **Requisiti non funzionali**
     1. **Usabilità**

Il sistema è stato progettato pensando principalmente al problema che i futuri utilizzatori non saranno, almeno non tutti, “addetti ai lavori” quindi maggior attenzione è stata rivolta alla facilità d’utilizzo, comprendendo interfacce semplificate

* + 1. **Affidabilità**

Il sistema deve consentire l’accesso ad informazioni corrette, sempre aggiornate ed allineate tra di loro. Tutte le operazione sono quindi progettate per essere robuste ed evitare incongruenze nel database in caso di eccezioni

* + 1. **Prestazioni**

I tempi di operazione devono essere minimi e verrà quindi scelto un server capace di rispondere velocemente alle richieste dell’utente.

* + 1. **Supportabilità**

La manutenibilità del sistema può essere svolta in maniera semplice e l’applicazione può essere estesa per sviluppi futuri, es. aggiunta nuove funzionalità.

* 1. **Modelli del sistema**
     1. **Scenari**

Si prevedono due tipi di attori: Amministratori(uno o più di uno) e utenti

* + - 1. **Scenario gestione leghe**

|  |  |
| --- | --- |
| **Nome scenario:** Creazione lega | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore accede alla pagina effettuando il login. 2. L’amministratore deve creare una nuova lega. 3. Accede, tramite il pannello di controllo, alla sezione “crea lega”. 4. Il sistema presenta il modulo per la creazione della lega. 5. L’amministratore compila il form. 6. L’amministratore controlla i dati inseriti e conferma. |

|  |  |
| --- | --- |
| **Nome scenario:** Cancellazione lega | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore accede alla pagina effettuando il login. 2. L’amministratore deve cancellare una lega perché in disuso. 3. Accede, tramite il pannello di controllo, alla sezione “elimina lega”. 4. Il sistema chiede conferma per l’eliminazione della lega. 5. L’amministratore conferma l’eliminazione. |

|  |  |
| --- | --- |
| **Nome scenario:** Accorpamento leghe | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore accede alla pagina effettuando il login. 2. L’amministratore deve accorpare una lega con pochi iscritti ad un’altra. 3. Accede, tramite il pannello di controllo, alla sezione “accorpa leghe”. 4. Il sistema chiede all’amministratore in quale lega devono essere trasferiti gli utenti. 5. L’amministratore seleziona la lega dove verranno trasferiti gli utenti. 6. L’amministratore conferma. 7. Il sistema chiede all’amministratore se eliminare la lega ormai vuota. 8. L’amministratore elimina la lega vuota. |

* + - 1. **Gestione utenti**

|  |  |
| --- | --- |
| **Nome scenario:** Accettare un utente | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore riceve una richiesta di partecipazione ad una lega da parte di un utente. 2. L’amministratore accede alla pagina effettuando il login. 3. Accede, tramite il pannello di controllo, alla sezione “gestione utenti”. 4. Controlla i dati. 5. Verifica l’utente. 6. Inserisce l’utente nella lega. |

|  |  |
| --- | --- |
| **Nome scenario:** Bannare un utente | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore riceve diverse notifiche sul comportamento scorretto di un utente. 2. L’amministratore accede alla pagina effettuando il login. 3. Accede, tramite il pannello di controllo, alla sezione “gestione utenti”. 4. Verifica la veridicità delle notifiche. 5. Banna l’utente in questione. |

|  |  |
| --- | --- |
| **Nome scenario:** Eliminare una account utente in disuso | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore accede alla pagina effettuando il login. 2. Accede, tramite il pannello di controllo, alla sezione “gestione utenti”. 3. L’amministratore controlla se un account non è usato da molto tempo. 4. Manda una mail di verifica se l’utente è ancora attivo. 5. Se il riscontro è negativo elimina l’utente. |

* + - 1. **Gestione voti**

|  |  |
| --- | --- |
| **Nome scenario:** Inserimento voti | |
| **Partecipanti** | Amministratore |
| **Flusso di eventi** | 1. L’amministratore accede alla pagina effettuando il login. 2. Accede, tramite il pannello di controllo, alla sezione “upload voti” 3. Inserisce il file con i voti. 4. Conferma l’upload. 5. Invia un messaggio in broadcast agli utenti che i voti sono online. |

* + 1. **Use case**

**Use case utente generico: Amministratore, utente**

|  |  |
| --- | --- |
| **Nome use case:** Login | |
| **Partecipanti** | Utente generico |
| **Condizione di ingresso** | L’utente generico accede al sistema  L’utente è registrato |
| **Flusso di eventi** | 1. L’utente generico clicca su accedi. 2. Il sistema riceve la scelta fatta dall’utente generico e visualizza il form. 3. L’utente generico inserisce username e password e sottomette il form cliccando su login. 4. Il sistema riceve ed elebora i dati e notifica il successo o l’insuccesso dell’operazione. |
| **Eccezioni** | Dati non presenti  Errore nel formato dati |
| **Condizioni di uscita** | L’utente ha avuto accesso al sistema |

|  |  |
| --- | --- |
| **Nome use case:** Logout | |
| **Partecipanti** | Utente generico |
| **Condizione di ingresso** | L’utente generico è loggato al sistema |
| **Flusso di eventi** | 1. L’utente generico clicca su logout 2. Il sistema disconnette l’utente |
| **Eccezioni** | Il sistema non riesce a disconnettere l’utente |
| **Condizioni di uscita** | L’utente ha effettuato il logout |

|  |  |
| --- | --- |
| **Nome use case:** Ricerca squadra | |
| **Partecipanti** | Utente generico |
| **Condizione di ingresso** | L’utente generico accede al sistema |
| **Flusso di eventi** | 1. L’utente immette il nome della squadra nella barra di ricerca 2. L’utente conferma 3. Il sistema mostra i dati |
| **Eccezioni** | Dati non presenti  Errore nel formato dati |
| **Condizioni di uscita** | Vengono restituiti uno o più risultati |

|  |  |
| --- | --- |
| **Nome use case:** Ricerca utente | |
| **Partecipanti** | Utente generico |
| **Condizione di ingresso** | L’utente generico accede al sistema |
| **Flusso di eventi** | 1. L’utente immette il nome dell’utente nella barra di ricerca 2. L’utente conferma 3. Il sistema mostra i dati |
| **Eccezioni** | Dati non presenti  Errore nel formato dati |
| **Condizioni di uscita** | Vengono restituiti uno o più risultati |

**Use case Amministratore**

|  |  |
| --- | --- |
| **Nome use case:** Creazione lega | |
| **Partecipanti** | Amministratore |
| **Condizione di ingresso** | L’Amministratore accede al sistema |
| **Flusso di eventi** |  |
| **Eccezioni** |  |
| **Condizioni di uscita** |  |