|  |
| --- |
| Progetto di Ingegneria del Software 2015/16  U_F_L.jpg  **U.F.L.**  University Fanta League   * **Object Design 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 |
| 15/12/2015 | 1.0 | Stesura del testo | Stefano Foresta |
|  | 1.0 |  | Gennaro Franzese |
|  | 1.0 |  | Giuseppe Paglialonga |

*Indice*

1. **Introduzione**

Lo scopo del documento di Object Design è quello di specificare i servizi che ogni sottosistema, presentato nel documento di System Design, offre in termini di delle classi includendo operazioni, tipi, argomenti e signature in modo da avere una specifica completa. Questo documento serve come base per l’implementazione.

* 1. **Scelte nel Design degli Oggetti**
     1. **Interfaccia vs. Usabilità**

Un’interfaccia grafica semplice e chiara che servirà a rendere usabile ed intuitivo il prodotto che verrà realizzato.

* + 1. **Tempo di rilascio vs. Tolleranza ai fault**

Per la tolleranza di eventuali fault si procederà con dei backup mensili del database che contiene i dati che rendono possibile il funzionamento della web application.

* + 1. **Sicurezza vs Efficienza**

Si è prevista l’implementazione di un modulo di autenticazione che renderà più sicuro il software attraverso politiche di gestione delle password.

* + 1. **Comprensibilità vs Tempo**

Si cercherà, attraverso la suddivisione del codice e l’introduzione di commenti, di rendere il codice comprensibile anche a un terzo che non ha partecipato al progetto.

1. **Package**

Il Software ha un’architettura two-tier. Possiamo comunque distinguere tre differenti livelli:

* Interface Layer
* Logic Application Layer
* Storage Layer

I layer sono suddivisi in package ed ognuno di questi contiene degli oggetti che vanno a implementare le diverse funzionalità del sistema sviluppato. I package vengono usati per riunire le classi logicamente correlate e che forniscono servizi simili.

* 1. **Descrizione dei layer**

*Interface Layer*

|  |  |
| --- | --- |
| MODULO | DESCRIZIONE |
| **Interfaccia web** | Modulo che rappresenta l’interfaccia grafica web con cui l’utente interagirà con il sistema inoltrando le richieste. |

*Logic application layer*

|  |  |
| --- | --- |
| **MODULO** | **DESCRIZIONE** |
| **Gestione Utente** | Modulo che si occupa della creazione, modifica e  eliminazione dell’account utente. |
| **Gestione Leghe** | Modulo che si occupa della gestione delle leghe,inserimento squadre, aggiornamento classifica |
| **Gestione Giornata** | Modulo che si occupa delle formazioni dei giocatori dell’utente, dell’inserimento dei voti da parte dell’amministratore e del calcolo e aggiornamento delle classifiche. |
| **Gestione squadra** | Modulo che permette all’utente di realizzare e aggiornare la rosa dei giocatori della squadra |
| **Gestione mercato** | Modulo che permette la gestione delle modifiche alle squadre, consente o meno la modifica delle squadre da parte degli utenti, aggiunge giocatori al database, modifica i valori dei giocatori |
| **Informazioni** | Modulo che permette agli utenti di visualizzare informazioni sugli utenti iscritti , le squadre e le classifiche |
| **Storage** | Modulo che si occupa della comunicazione con il database |
|  | |

*Storage application*

|  |  |
| --- | --- |
| **Modulo** | **Descrizione** |
| **Database** | Modulo che inoltra le richieste di recupero dati direttamente al database fisico e ne recupera i dati, per poi restituirli allo strato delle applicazioni logiche. |

* 1. **Comunicazione tra package**

Il seguente diagramma mostra la comunicazione tra i package che compongono i vari layer del sistema. La scelta progettuale è quella di far passare ogni richiesta fatta dalle classi di interfaccia al Logic Application Layer. All’interno del Logic Application Layer sono presenti i pacchetti che si occupano della logica applicativa del sistema e della sua gestione. Infine il Logic Application Layer accederà, attraverso l'interfaccia Storage, allo Storage Layer, per recuperare, aggiornare o eliminare dati persistenti dalla base dati.

****

1. **Class Interfaces**

* Ogni classe ha necessariamente un nome che è significativo per il suo scopo.
* Ogni classe ha un nome che è diverso da tutte le altre classi.
  1. **Gestione utente**

Modulo che si occupa della creazione, modifica e eliminazione dell’account utente.

* + 1. Creazione account

Dipendenze: nessuna dipendenza.

Eccezioni: utente già nel database, dati non validi.

* + 1. Modifica account

Dipendenze: ricerca.

Eccezioni: utente già presente, dati non validi.

* + 1. Eliminazione account

Dipendenze: ricerca.

Eccezioni: nessuna eccezione.

* 1. **Gestione leghe**

Modulo che si occupa della gestione delle leghe,inserimento squadre, aggiornamento classifica.

* + 1. Avvia iscrizioni

Dipendenze: database.

Eccezioni: database non popolato.

* + 1. Carica calendario

Dipendenze: nessuna.

Eccezioni: file già presente, formato file errato.

* + 1. Popola database

Dipendenze: nessuna.

Eccezioni: formato errato.

* 1. **Giornata**

Modulo che si occupa delle formazioni dei giocatori dell’utente, dell’inserimento dei voti da parte dell’amministratore e del calcolo e aggiornamento delle classifiche.

* + 1. Visualizza classifica
    2. Visualizza risultati giornata
  1. **Gestione squadra**

Modulo che permette all’utente di realizzare e aggiornare la rosa dei giocatori della squadra.

* + 1. Realizzazione formazione

Dipendenze: ricerca.

Eccezioni: formazione non completa.

* 1. **Gestione mercato**

Modulo che permette la gestione delle modifiche alle squadre, consente o meno la modifica delle squadre da parte degli utenti, aggiunge giocatori al database, modifica i valori dei giocatori.

* + 1. Composizione rosa

Dipendenze: ricerca.

Eccezioni: giocatore già presente, fantamilioni non disponibili.

* + 1. Modifica rosa

Dipendenze: ricerca.

Eccezioni: giocatore già presente, fantamilioni non disponibili, rosa non modificabile.

* 1. **Informazioni**

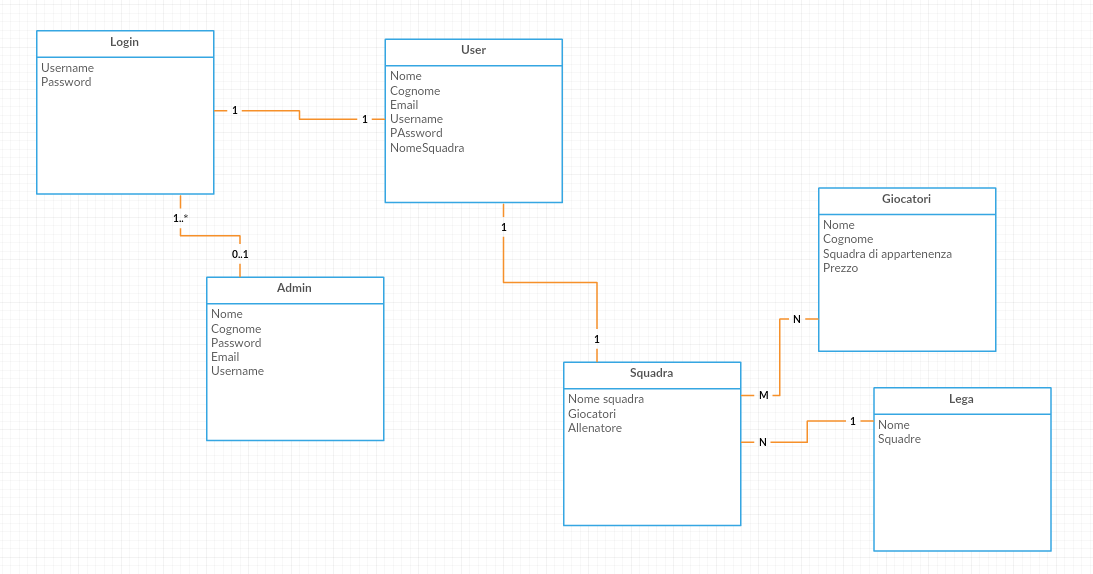
Modulo che permette agli utenti di visualizzare informazioni sugli utenti iscritti, le squadre e le classifiche.

* 1. **Storage**

Modulo che si occupa della comunicazione con il database.

* 1. **Calendario**

Modulo che permette la visualizzazione del campionato in merito agli incontri che sono stati calendarizzati in precedenza.

* 1. **Interfaccia**
     1. Form login.
     2. Visualizzaione menù.
  2. **Class Diagram**