



Summer Center

**Object Design Document**

# Top manager

| **Professore** | **Tutor** |
| --- | --- |
| De Lucia Andrea | Pecorelli Fabiano |

# Partecipanti

| **Cognome** | **Nome** | **Matricola** |
| --- | --- | --- |
| **Dalia** | Antonio | 0512106118 |
| **Donnarumma** | Gerardo | 0512106064 |
| **Esposito** | Vincenzo | 0512106070 |

# Storico versioni

| **Data** | **Versione** | **Descrizione** | **Autori** |
| --- | --- | --- | --- |
| **31/01/2021** | 1.0 | Creazione del documento, dell’introduzione e dei package | Esposito |
| **01/02/2021** | 1.1 | Interfaccia delle classi: Utente, Genitore | Esposito |

Sommario

[Top manager 2](#_Toc63166639)

[Partecipanti 2](#_Toc63166640)

[Storico versioni 2](#_Toc63166641)

[Introduzione 4](#_Toc63166642)

[Trade-offs di Object Design 4](#_Toc63166643)

[**Funzionalità vs Usabilità** 4](#_Toc63166644)

[**Sviluppo rapido vs Funzionalità** 4](#_Toc63166645)

[**Manutenibilità vs Performance** 4](#_Toc63166646)

[Linee guida per la documentazione dell’interfaccia 4](#_Toc63166647)

[Definizioni, acronimi e abbreviazioni 4](#_Toc63166648)

[Packages 5](#_Toc63166649)

[view 5](#_Toc63166650)

[control.gestioneAutenticazione 5](#_Toc63166651)

[control.gestioneIscrizione 5](#_Toc63166652)

[model.entity 6](#_Toc63166653)

[model.dao 6](#_Toc63166654)

[Interfaccia delle classi 7](#_Toc63166655)

[Utente 7](#_Toc63166656)

[Genitore 8](#_Toc63166657)

[Bambino 10](#_Toc63166658)

# Introduzione

## Trade-offs di Object Design

### **Funzionalità vs Usabilità**

Abbiamo deciso di preferire l’usabilità laddove fosse necessario in quanto l’uso della web-application deve essere il più intuitivo possibile per l’utente finale.

Le funzionalità non implementate potranno essere successivamente sviluppate e attivate in maniera graduale.

### **Sviluppo rapido vs Funzionalità**

Abbiamo deciso di non implementare alcune funzionalità a causa delle scadenze fissate e di preferire lo sviluppo e il testing delle funzionalità ad alta priorità per rendere più fruibile il sistema per l’iscrizione ed eventualmente il pagamento.

### **Manutenibilità vs Performance**

Abbiamo preferito le performance per quanto riguarda la gestione dei dati persistenti. Infatti, i file delle Iscrizioni saranno memorizzati direttamente sul disco rigido.

## Linee guida per la documentazione dell’interfaccia

Affinché il codice sia facilmente leggibile e mantenibile, si è deciso di adottare la notazione [CamelCase](https://en.wikipedia.org/wiki/Camel_case) in quanto è la più familiare per gli sviluppatori coinvolti.

Esempio di nome per una variabile: nomeVariabile

Esempio di nome per una classe: NomeClasse

È consigliato l’uso della seguente forma d’uso delle parentesi graffe per la scrittura di metodi, classi, cicli ed istruzioni di controllo:

Nomecampo{

//Codice

}

Il codice dovrà essere opportunamente indentato e l’ambiente di sviluppo Eclipse permette l’indentazione delle istruzioni.

## Definizioni, acronimi e abbreviazioni

|  |  |
| --- | --- |
| Denominazione | Formato |
| Random Control | Le classi Control gestiscono le richieste HTTP e inviano le risposte al client |
| Random manager | Le classi Manager gestiscono la Business Logic e l’interazione con i file e il database. |
| Jsp | [JavaServer Pages](https://www.oracle.com/java/technologies/jspt.html) |

# Packages

## view

|  |  |
| --- | --- |
| Denominazione | Formato |
| index.jsp | Home del sistema |
| Genitore/homegenitore.jsp | Home del genitore |
| Genitore/iscrizione.jsp | Pagina del form per l’iscrizione del bambino |
| Genitore/pagamento.jsp | Pagina per effettuare il pagamento delle iscrizioni |
| Genitore/visualizzaIscrizioni.jsp | Pagina che mostra tutte le iscrizioni effettuate (pagate e non) |
| Login.jsp | Pagina che permette di effettuare il login al sistema |

## control.gestioneAutenticazione

|  |  |
| --- | --- |
| Denominazione | Formato |
| LoginControl | Gestisce il login |
| LogoutControl | Gestisce il logout |

## control.gestioneIscrizione

|  |  |
| --- | --- |
| Denominazione | Formato |
| IscrizioneControl | Gestisce i campi inseriti per l’iscrizione di un bambino e ne controlla il formato |
| SettimaneControl | Gestisce le settimane per le iscrizioni |
| BambinoControl | Gestisce i campi del bambino e ne controlla il formato |

## model.entity

|  |  |
| --- | --- |
| Denominazione | Formato |
| Utente | Rappresenta un utente |
| Genitore | Rappresenta un genitore |
| Iscrizione | Rappresenta un’iscrizione |
| Bambino | Rappresenta un bambino |

## model.dao

|  |  |
| --- | --- |
| Denominazione | Formato |
| UtenteManager | Modella specifiche interazioni degli utenti con il DB |
| GenitoreManager | Modella le interazioni del Genitore con il DB |
| IscrizioneManager | Modelle le interazioni delle Iscrizioni con il DB |
| BambinoManager | Modella le interazioni del Bambino con il DB |

# Interfaccia delle classi

## Utente

|  |  |
| --- | --- |
| Nome | Utente |
| Descrizione | Rappresenta un utente registrato nel sistema |
| Attributi | -nome: String  -cognome: String  -email: String  -codiceFiscale: String  -dataDiNascita: Date  -età: integer  -password: String  -ruolo: String |
| Signature dei metodi | +getNome()  +setNome(String nome)  +getCognome()  +setCognome(String cognome)  +getEmail()  +setEmail(String email)  +getCF()  +setCF(String codiceFiscale)  +getDataDiNascita()  +setDataDiNascita(Date dataDiNascita)  +getEtà()  +setEtà(int età)  +getPsw()  +setPsw (String password) |
| Pre-condizioni | **context** Utente::setPassword(p) **pre**:  p.matches(‘^(?=.\*[0-9])(?=.\*[a-zA-Z])([a-zA-Z0-9]+){8,64}$’) |
| **context** Utente::setEmail(e) **pre**:  e.matches(‘\S+@\S+\.\S+’) |
| **context** Utente::setUsername(u) **pre**:  u.matches(‘^[A-Za-z0-9]{​​6,25}​​$’) |
| **context** Utente::setCellulare(c) **pre**:  c.matches(‘^[0-9]{8,16}$’) |
| **context** Utente::setTelefono(t) **pre**:  t.matches(‘^[0-9]{8,16}$’) |
|  |
| Post-condizioni |  |
| Invarianti |  |

## Genitore

|  |  |
| --- | --- |
| Nome | Genitore |
| Descrizione | Rappresenta un genitore registrato nel sistema |
| Attributi | -luogoNascita: String  -indirizzo: String  -città: String  -cap: Intenger  -professione: String  -nTel1: integer  -nTel2: integer  -documentoIdentità:  -consensoInfo: Boolean  -consensoImgVideo: Boolean  -delega: String  -infoFamiliari: String |
| Signature dei metodi | +getLuognoNascita()  +setLuogoNascita(String luogoNascita)  +getIndirizzo()  +setIndirizzo(String indirizzo)  +getCittà()  +setCittà(String città)  +getCap()  +setCap(int cap)  +getProfessione()  +setProfessione(String professione)  +getNTel1()  +setNTel1(int nTel1)  +getNTel2()  +setNTel1(int nTel2)  +getDocumentoIdentità()  +setDocumentoIdentità(… documentoIdentità)  +getConsensoInfo()  +setConsensoInfo(Boolean consensoInfo)  +getConsensoImgVideo()  +setConsensoImgVideo(Boolean consensoImgVideo)  +getDelega()  +setDelega(String delega)  +getInfoFamiliari()  +setInfoFamiliari (String infoFamiliari) |
| Pre-condizioni | **context** Utente::setPassword(p) **pre**:  p.matches(‘^(?=.\*[0-9])(?=.\*[a-zA-Z])([a-zA-Z0-9]+){8,64}$’) |
| **context** Utente::setUsername(u) **pre**:  u.matches(‘^[A-Za-z0-9]{​​6,25}​​$’) |
| **context** Utente::setCellulare(c) **pre**:  c.matches(‘^[0-9]{8,16}$’) |
| **context** Utente::setTelefono(t) **pre**:  t.matches(‘^[0-9]{8,16}$’) |
| **context** Utente::setEmail(e) **pre**:  e.matches(‘\S+@\S+\.\S+’) |
| Post-condizioni |  |
| Invarianti |  |

## Bambino

|  |  |
| --- | --- |
| Nome | Bambino |
| Descrizione | Rappresenta un bambino iscritto nel sistema |
| Attributi | -nome: String  -cognome: String  -dataDiNascita: Date  -luogoNascita: String  -codiceFiscale: String  -genere: String  -esigenzeAlimentari: Boolean  -disabilità: Boolean  -descrizioneEsigenzeAlimentari: String  -materialeGalleggiante: Boolean  -farmaci: String  -allergie: String  -documentoIdentità: String  -certificatoMedico: String |
| Signature dei metodi | +getNome()  +setNome(String nome)  +getCognome()  +setCognome(String cognome)  +getDataDiNasciata()  +setDataDiNasciata(Date dataDiNascita)  +getLuognoNascita()  +setLuogoNascita(String luogoNascita)  +getCF()  +setCF(String codiceFiscale)  +getGenere()  +setGenere(String genere)  +getEsigenzeAlimentari()  +setEsigenzeAlimentari(Boolean esigenzeAlimentari)  +getDisabilità()  +setDisabilità(Boolean disabilità)  +getDescrizioneEsigenzeAlimentari()  +setDescrizioneEsigenzeAlimentari(String descrizioneEsigenzeAlimentari)  +getMaterialeGalleggiante()  +setMaterialeGalleggiante(Boolean materialeGalleggiante)  +getFarmaci()  +setFarmaci(String farmaci)  +getAllergie()  +setAllergie(String allergie)  +getDocumentoIdentità()  +setDocumentoIdentità(String documentoIdentità)  +getCertificatoMedico ()  +setCertificatoMedico (String certificatoMedico) |
| Pre-condizioni | **context** Utente::setPassword(p) **pre**:  p.matches(‘^(?=.\*[0-9])(?=.\*[a-zA-Z])([a-zA-Z0-9]+){8,64}$’) |
| **context** Utente::setUsername(u) **pre**:  u.matches(‘^[A-Za-z0-9]{​​6,25}​​$’) |
| **context** Utente::setCellulare(c) **pre**:  c.matches(‘^[0-9]{8,16}$’) |
| **context** Utente::setTelefono(t) **pre**:  t.matches(‘^[0-9]{8,16}$’) |
| **context** Utente::setEmail(e) **pre**:  e.matches(‘\S+@\S+\.\S+’) |
| Post-condizioni |  |
| Invarianti |  |