



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 |
| **08/02/2021** | 1.2 | Interfaccia delle classi: Bambino, Settimana, Iscrizione | Esposito |
| **09/02/2021** | 1.3 | Interfaccia delle classi: UtenteManage, IscrizioneManage, SettimanaManage | Esposito |
| **10/02/2021** | 1.4 | Aggiunta dello schema E-R del Database | Esposito |

Sommario

[Top manager 2](#_Toc63802912)

[Partecipanti 2](#_Toc63802913)

[Storico versioni 2](#_Toc63802914)

[Introduzione 7](#_Toc63802915)

[Trade-offs di Object Design 7](#_Toc63802916)

[**Funzionalità vs Usabilità** 7](#_Toc63802917)

[**Sviluppo rapido vs Funzionalità** 7](#_Toc63802918)

[**Manutenibilità vs Performance** 7](#_Toc63802919)

[Linee guida per la documentazione dell’interfaccia 7](#_Toc63802920)

[Definizioni, acronimi e abbreviazioni 7](#_Toc63802921)

[Schema E-R 8](#_Toc63802922)

[Packages 9](#_Toc63802924)

[view 9](#_Toc63802925)

[control.gestioneAutenticazione 9](#_Toc63802926)

[control.gestioneIscrizione 9](#_Toc63802927)

[model.entity 10](#_Toc63802928)

[model.dao 10](#_Toc63802929)

[Interfaccia delle classi 11](#_Toc63802930)

[Utente 11](#_Toc63802931)

[Genitore 13](#_Toc63802932)

[Bambino 15](#_Toc63802933)

[Iscrizione 17](#_Toc63802934)

[Settimana 19](#_Toc63802935)

[UtenteManage 20](#_Toc63802936)

[BambinoManage 21](#_Toc63802937)

[IscrizioneManage 22](#_Toc63802938)

[SettimanaManage 23](#_Toc63802939)

# 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

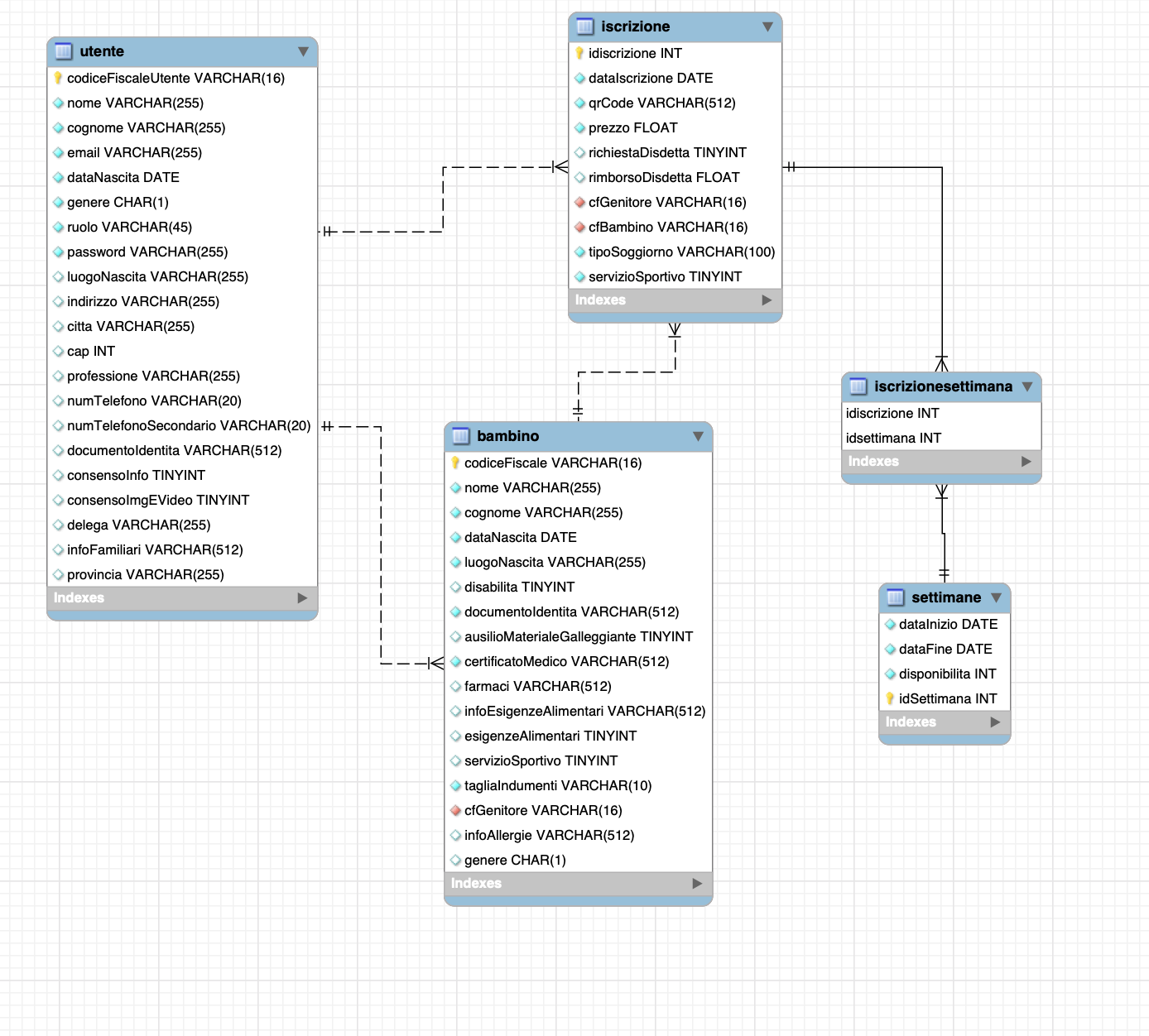
}

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) |

# Schema E-R



# 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 |
| Settimana | Rappresenta una settimana |

## model.dao

|  |  |
| --- | --- |
| Denominazione | Formato |
| UtenteManage | Modella specifiche interazioni degli utenti con il DB |
| GenitoreManage | Modella le interazioni del Genitore con il DB |
| IscrizioneManage | Modelle le interazioni delle Iscrizioni con il DB |
| BambinoManage | Modella le interazioni del Bambino con il DB |
| settimanaManage | Modella le iterazioni del Settimane 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  -genere: char  -password: String  -ruolo: String  -numTelefono: String |
| Signature dei metodi | +getNome()  +setNome(String nome)  +getCognome()  +setCognome(String cognome)  +getEmail()  +setEmail(String email)  +getCF()  +setCF(String codiceFiscale)  +getDataDiNascita()  +setDataDiNascita(Date dataDiNascita)  +getRuolo()  +setRuolo(char ruolo)  +getPassword()  +setPassword (String password)  +getNumTelefono()  +setNumTelfono(String numTelefono) |
| Pre-condizioni | **context** Utente::setPassword(p) **pre**:  p.matches(‘^(?=.\*[0-9])(?=.\*[a-zA-Z])([a-zA-Z0-9]+){8,64}$’) |
| **context** Utente::setNome(n) **pre**:  n.matches(‘^[a-zA-Z]+$’) |
| **context** Utente::setCognome(c) **pre**:  c.matches(‘^[a-zA-Z]+$’) |
| **context** Utente::setEmail(e) **pre**:  e.matches(‘\S+@\S+\.\S+’) |
| **context** Utente::setNumTelefono(t) **pre**:  t.matches(‘^[0-9]{8,16}$’) |
| **context** Utente::setCodiceFiscale(cf) **pre**:  cf.matches(‘^[a-zA-Z]{6}[0-9]{2}[abcdehlmprstABCDEHLMPRST]{1}[0-9]{2}([a-zA-Z]{1}[0-9]{3})[a-zA-Z]{1}$’) |
| **context** Utente::setDataNascita(dataNascita) **pre**:  dataNascita.lenght>8 and dataNascita.lenght<16 and dataNascita.matches(‘^(0?[1-9]|[12][0-9]|3[01])[/](0?[1-9]|1[012])[/]\d{4}$’) and dataNascita<now() |
| Post-condizioni |  |
| Invarianti |  |

## Genitore

|  |  |
| --- | --- |
| Nome | Genitore |
| Descrizione | Rappresenta un genitore registrato nel sistema |
| Attributi | -luogoNascita: String  -indirizzo: String  -città: String  -provincia: String  -cap: Intenger  -professione: String  -numTelefonoSecondario: String  -documentoIdentita: String  -consensoInfo: Boolean  -consensoImgVideo: Boolean  -delega: String  -infoFamiliari: String |
| Signature dei metodi | +getLuognoNascita()  +setLuogoNascita(String luogoNascita)  +getIndirizzo()  +setIndirizzo(String indirizzo)  +getCittà()  +setCittà(String città)  +getProvincia()  +setProvincia(String provincia)  +getCap()  +setCap(int cap)  +getProfessione()  +setProfessione(String professione)  +getNumTelefonoSecondario()  +setNumTelefonoSecondario(String numTelefonoSecondario)  +getDocumentoIdentita()  +setDocumentoIdentita(String documentoIdentita)  +getConsensoInfo()  +setConsensoInfo(Boolean consensoInfo)  +getConsensoImgVideo()  +setConsensoImgVideo(Boolean consensoImgVideo)  +getDelega()  +setDelega(String delega)  +getInfoFamiliari()  +setInfoFamiliari (String infoFamiliari) |
| Pre-condizioni | **context** Genitore::setLuogoNascita(ln) **pre**:  ln.matches(‘^[a-zA-Z0-9,]\*$’) |
| **context** Genitore::setIndirizzo(i) **pre**:  i.matches(‘^[a-zA-Z0-9,]\*$’) |
| **context** Genitore::setCitta(c) **pre**:  c.matches(‘^[a-zA-Z]+$’) |
| **context** Genitore::setProvincia(p) **pre**:  p.matches(‘^[a-zA-Z]+$’) |
| **context** Genitore:: setCap(a) **pre**:  a.length = 5 and a.matches(‘^[0-9]{5}$’) |
| **context** Genitore::setProfessione (p) **pre**:  p.matches(‘^[a-zA-Z]+$’) |
| **context** Genitore::setDelega (d) **pre**:  d.matches(‘^[a-zA-Z]+$’) |
| **context** Genitore::setInfoFamiliari(info) **pre**:  info.matches(‘^[a-zA-Z]+$’) |
| **context** Genitore::setNumTelefonoSecondario(t) **pre**:  t.matches(‘^[0-9]{8,16}$’) |
| **context** Genitore::setNumTelefonoSecondario(t) **pre**:  t.matches(‘^[0-9]{8,16}$’) |
| 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  -eta: Integer  -esigenzeAlimentari: Boolean  -disabilità: Boolean  -infoEsigenzeAlimentari: String  -ausilioMaterialeGalleggiante: 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** Bambino::setNome(n) **pre**:  n.matches(‘^[a-zA-Z]+$’) |
| **context** Bambino::setCognome(c) **pre**:  c.matches(‘^[a-zA-Z]+$’) |
| **context** Bambino::setDataNascita(dataNascita) **pre**:  dataNascita.lenght>8 and dataNascita.lenght<16 and dataNascita.matches(‘^(0?[1-9]|[12][0-9]|3[01])[/](0?[1-9]|1[012])[/]\d{4}$’) and dataNascita<now() |
| **context** Bambino::setCodiceFiscale(cf) **pre**:  cf.matches(‘^[a-zA-Z]{6}[0-9]{2}[abcdehlmprstABCDEHLMPRST]{1}[0-9]{2}([a-zA-Z]{1}[0-9]{3})[a-zA-Z]{1}$’) |
| **context** Bambino::setLFarmaci(f) **pre**:  f.matches(‘^[a-zA-Z0-9,]\*$’) |
| **context** Bambino::setAllergie(a) **pre**:  a.matches(‘^[a-zA-Z]+$’) |
| **context** Bambino::setInfoEsigenzeAlimentari(info) **pre**:  info.matches(‘^[a-zA-Z]+$’) |
| **context** Bambino::setLuogoNascita(ln) **pre**:  ln.matches(‘^[a-zA-Z0-9,]\*$’) |
| Post-condizioni |  |
| Invarianti |  |

## Iscrizione

|  |  |
| --- | --- |
| Nome | Iscrizione |
| Descrizione | Rappresenta un’iscrizione nel sistema |
| Attributi | -idIscrizione: Integer  -dataIscrizione: Date  -qrCode: String  -prezzo: float  -richiestaDisdetta: Boolean  -rimborsoDisdetta: float  -servizioSportivo: Boolean  -tipoSoggiorno: String  - |
| Signature dei metodi | +getIdIscrizione()  +setIdIscrizione(int idIscrizione)  +getDataIscrizione()  +setDataIscrizione(Date dataIscrizione)  +getQrCode()  +setQrCode(String qrCode)  +getPrezzo()  +setPrezzo(float prezzo)  +getRichiestaRimborso()  +setRichiestaRimborso(Boolean richiestaRimborso)  +getRimborsoDisdetta ()  +setRimborsoDisdetta(float rimborsoDisdetta)  +getServizioSportivo()  +setServizioSportivo(Boolean servizioSportivo)  +getTipoSoggiorno()  +setTipoSoggiorno(String tipoSoggiorno) |
|  | **context** Settimana::setDataIscrizione(dataIscrizione) **pre**:  dataIscrizione.lenght>8 and dataIscrizione.lenght<16 and dataIscrizione.matches(‘^(0?[1-9]|[12][0-9]|3[01])[/](0?[1-9]|1[012])[/]\d{4}$’) and dataIscrizione<now() |
| **context** Settimana::setQrCode(qrCode) **pre**:  qrCode.matches(‘^[a-zA-Z0-9,]\*$’) |
| **context** Settimana::setPrezzo(p) **pre**:  p.matches(‘^[0-999,]\*$’) |
| **context** Settimana::setRimborsoDisdetta(r) **pre**:  r.matches(‘^[0-999,]\*$’) |
| **context** Settimana::setTipoSoggiorno(tipoSoggiorno) **pre**:  tipoSoggiorno.matches(‘^[a-zA-Z,]\*$’) |
| Post-condizioni |  |
| Invarianti |  |

## Settimana

|  |  |
| --- | --- |
| Nome | Settimana |
| Descrizione | Rappresenta una settimana nel sistema |
| Attributi | -idSettimana: Integer  -dataInizio: Date  -dataFine: Date  -disponibilita: Integer |
| Signature dei metodi | +getIdSettimana()  +setIdSettimana(int idSettimana)  +getDataInizio()  +setDataInzio(Date dataInizio)  +getDataFine()  +setDataFine(Date dataFine)  +getDisponibilita()  +setDisponibilita(int disponibilita) |
|  | **context** Settimana::setDataInizio(dataInizio) **pre**:  dataInizio.lenght>8 and dataInizio.lenght<16 and dataInzio.matches(‘^(0?[1-9]|[12][0-9]|3[01])[/](0?[1-9]|1[012])[/]\d{4}$’) |
| **context** Settimana::setDataFine(dataFine) **pre**:  dataFine.lenght>8 and dataFine.lenght<16 and dataFine.matches(‘^(0?[1-9]|[12][0-9]|3[01])[/](0?[1-9]|1[012])[/]\d{4}$’) and dataInizio<dataFine |
| Post-condizioni |  |
| Invarianti |  |

## UtenteManage

|  |  |
| --- | --- |
| Nome | UtenteManage |
| Descrizione | Modella le interazioni dell’Utente col Database |
| Attributi | -ds: DataSource  -TABLE\_NAME: String  -em: EntityManager |
| Signature dei metodi | -save(Utente u)  +getUserIfExists(String email, String password) |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invarianti |  |

## BambinoManage

|  |  |
| --- | --- |
| Nome | BambinoManage |
| Descrizione | Modella le interazioni del Bambino col Database |
| Attributi | -ds: DataSource  -TABLE\_NAME: String  -em: EntityManager |
| Signature dei metodi | +save(Bambino b);  +update(Bambino b);  +List<Bambino> getBambini(String cfGenitore);  +getBambino(String codiceFiscale); |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invarianti |  |

## IscrizioneManage

|  |  |
| --- | --- |
| Nome | IscrizioneManage |
| Descrizione | Modella le interazioni delle Iscrizioni col Database |
| Attributi | -ds: DataSource  -TABLE\_NAME: String  -em: EntityManager |
| Signature dei metodi | +save(Iscrizione i) |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invarianti |  |

## SettimanaManage

|  |  |
| --- | --- |
| Nome | SettimanaManage |
| Descrizione | Modella le interazioni della Settimana col Database |
| Attributi | -ds: DataSource  -TABLE\_NAME: String  -em: EntityManager |
| Signature dei metodi | +Settimana getSettimana(Settimana s)  +List<Settimana> getSettimaneDisponibili() |
| Pre-condizioni |  |
| Post-condizioni |  |
| Invarianti |  |