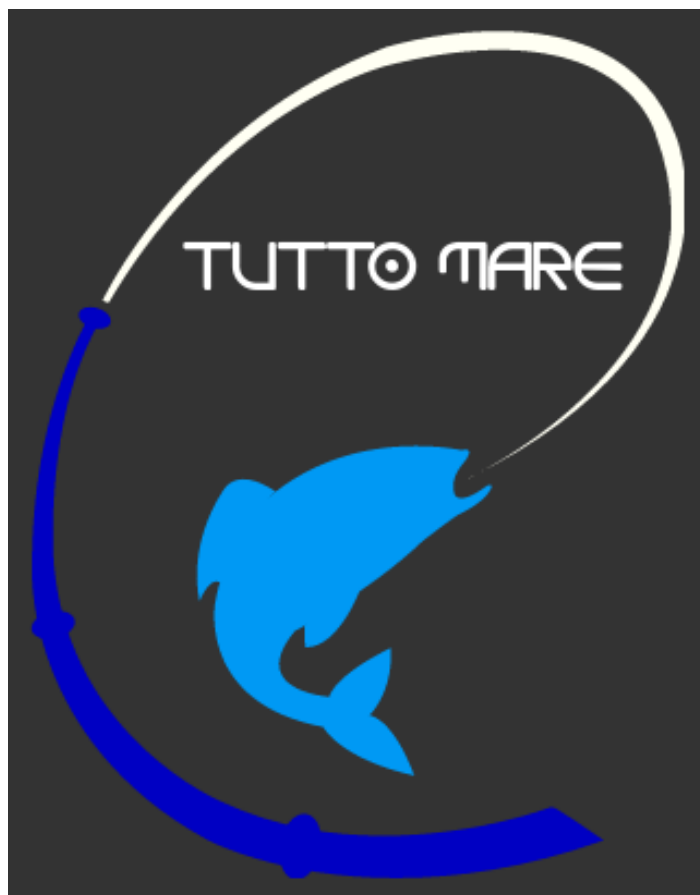


Object Design Document



Di:

Sara Guerriero 0512103956
Torre Francesca 0512102366

Indice degli Argomenti

1.0	Introduzione	3
1.1	Object design trade-offs	3
1.1.1	Comprensibilità e costi	3
1.1.2	Prestazione e costi	3
1.1.3	Costi e mantenimento	3
1.1.4	Interfaccia e easy-use	3
1.1.5	Memoria ed efficienza	3
1.1.6	Sicurezza e costi	3
1.1.7	Interfacce e tempo di risposta	3
1.2	Descrizione componenti off-the-shelf	4
1.3	Linee guida per la documentazione dell'interfaccia	4
1.4	Definizioni, acronimi e abbreviazioni	4
1.5	Riferimenti	5
2.0	Package	6
3.0	Interfacce delle classi	7
3.1	Gestione Account	9
3.2	Gestione Amministratore	10
3.3	Gestione ServiziUtente	11
4.0	Glossario	12

1.0 Introduzione

1.1 Object design trade-offs

1.1.1 Comprensibilità e costi

Vogliamo aggiungere i costi relativi alle ore di lavoro che l'uomo dedica alla documentazione del progetto, che saranno utili per comprendere al meglio le funzionalità del sistema. Nel codice saranno introdotti dei commenti per facilitarne sia la comprensione che la manutenzione.

1.1.2 Prestazione e costi

Il budget a disposizione non è eccessivo, e bisognerà quindi utilizzare componenti open source rendendo il sistema comunque soddisfacente.

1.1.3 Costi e mantenimento

Il sito si può facilmente modificare con nuove funzioni, ciò è possibile grazie all'uso di librerie open source.

1.1.4 Interfaccia e easy-use

L'interfaccia è di facile utilizzo ed è utilizzabile da qualsiasi tipo di utente.

1.1.5 Memoria ed efficienza

Dato l'elevato carico di utenti che andranno ad interfacciarsi sulla piattaforma, e dato che il sistema dovrà supportare un alto numero di query, per non penalizzare le performance del sistema dovremo utilizzare un meccanismo di caching che permette la memorizzazione dei dati. Ciò sarà fatto tramite l'utilizzo dei cookie.

1.1.6 Sicurezza e costi

Poichè il budget per il sistema è ridotto, non è garantita la massima sicurezza sui dati.

1.1.7 Interfacce e tempo di risposta

Il tempo di risposta tra server e interfaccia riesce a soddisfare le richieste da parte dell'utente.

1.2 Descrizione componenti off-the-shelf

Per il sito che vogliamo realizzare utilizziamo componenti off-the-shelf, che sono componenti software disponibili sul mercato per facilitare la creazione del progetto.

Per il sistema che si vuole realizzare faremo uso di un framework per la realizzazione dell'interfaccia grafica e di una libreria interna a Java per la crittografia dei dati.

Il framework scelto per l'interfaccia grafica è Bootstrap, un framework open source che contiene una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Questo contiene modelli di progettazione basati su HTML e CSS, per le varie componenti dell'interfaccia, come moduli, bottoni e navigazione, così come alcune estensioni opzionali di JavaScript.

1.3 Linee guida per la documentazione dell'interfaccia

Per rendere il codice più estensibile e manutenibile è consono sottomettere le regole di implementazione, in modo da correggere eventuali correzioni nella logica all'interno dell'applicazione.

Commenti: sono utili per dare una panoramica del codice e per fornire informazioni aggiuntive che non sono facilmente visibili nel codice.

Dichiarazioni

Posizionare le dichiarazioni all'inizio del blocco del codice.

Indentazione

L'indentazione deve essere effettuata con un TAB e ogni istruzione deve essere indentata.

Parentesi

A prescindere dalle istruzioni bisogna riportare il blocco di istruzioni tra parentesi graffe.

Script Javascript

Gli script che svolgono funzioni distinte dal funzionamento di una pagina, dovrebbero essere collocati in file separati.

1.4 Definizioni, acronimi e abbreviazioni

HTML: Linguaggio di mark-up per pagine web.

CSS: Linguaggio usato per definire la formattazione di pagine web.

Framework: Software di supporto allo sviluppo web.

Javascript: Linguaggio di scripting orientato agli oggetti e agli eventi

Eclipse: ambiente di sviluppo integrato multi-linguaggio e multi-piattaforma
Javadoc: è un applicativo incluso nel Java Development Kit utilizzato per la generazione automatica della documentazione del codice sorgente scritto in linguaggio Java.

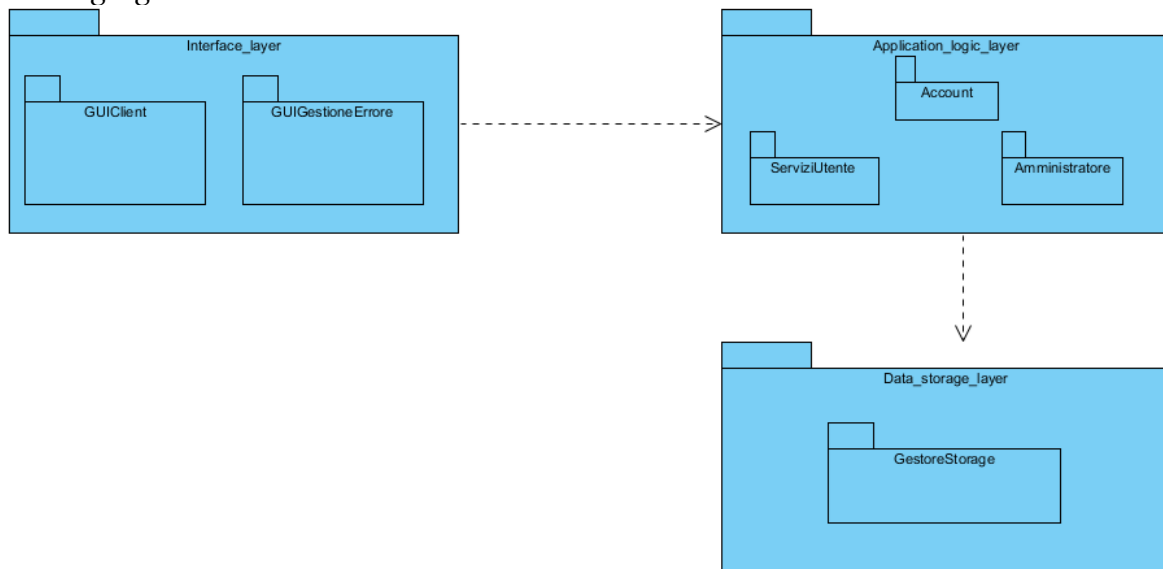
Off-The-Shelf: Servizi esterni di cui viene fatto utilizzo da terzi.

1.5 Riferimenti

- RAD – IMIS Documento di Analisi dei Requisiti.
- SDD – IMIS Documento di System Design
- Bernd Bruegge e Allen H. Dutoit - Object-Oriented Software Engineering (using UML, Patterns and JavaTM) – Prentice Hall
- Jim Conallen - Building Web Applications with UML (Second Edition) - Addison Wesley
- Guida completa JSP – Mc Graw Hill

2.0 Package

Package generale



Il diagramma descrive la natura three-layer dell'applicazione mostrandone i tre package principali:

InterfaceLayer: **GUIClient** e **GUIGestioneErrore**, indicano i sottosistemi che contengono tutti gli oggetti boundary;

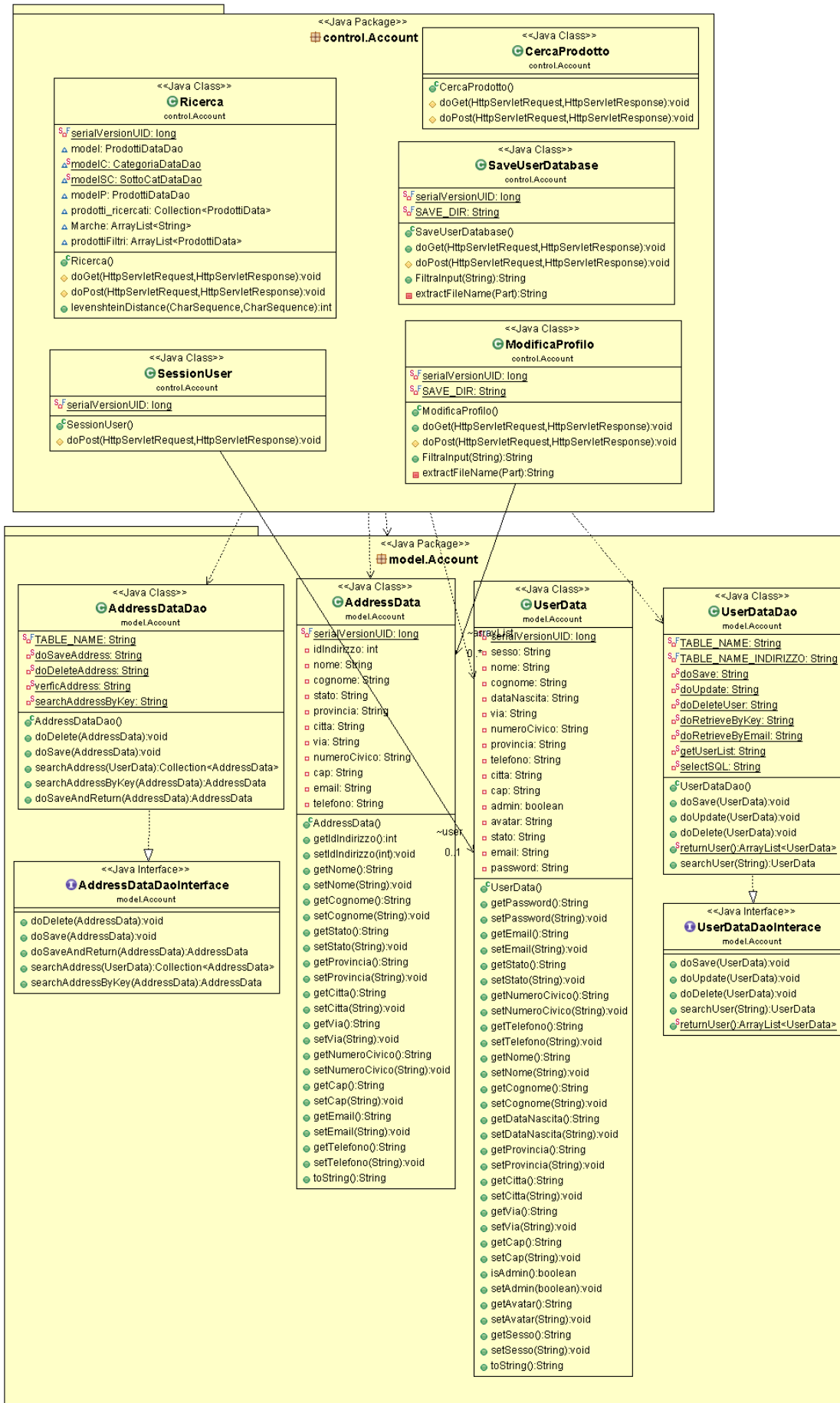
ApplicationLogicLayer: contiene i tre sottosistemi individuati (**Account**, **ServiziUtente**, **Amministratore**);

DataStorageLayer: **GestoreStorage**, sottosistema che ha il compito di effettuare operazioni verso il database.

3.0 Interfacce delle classi

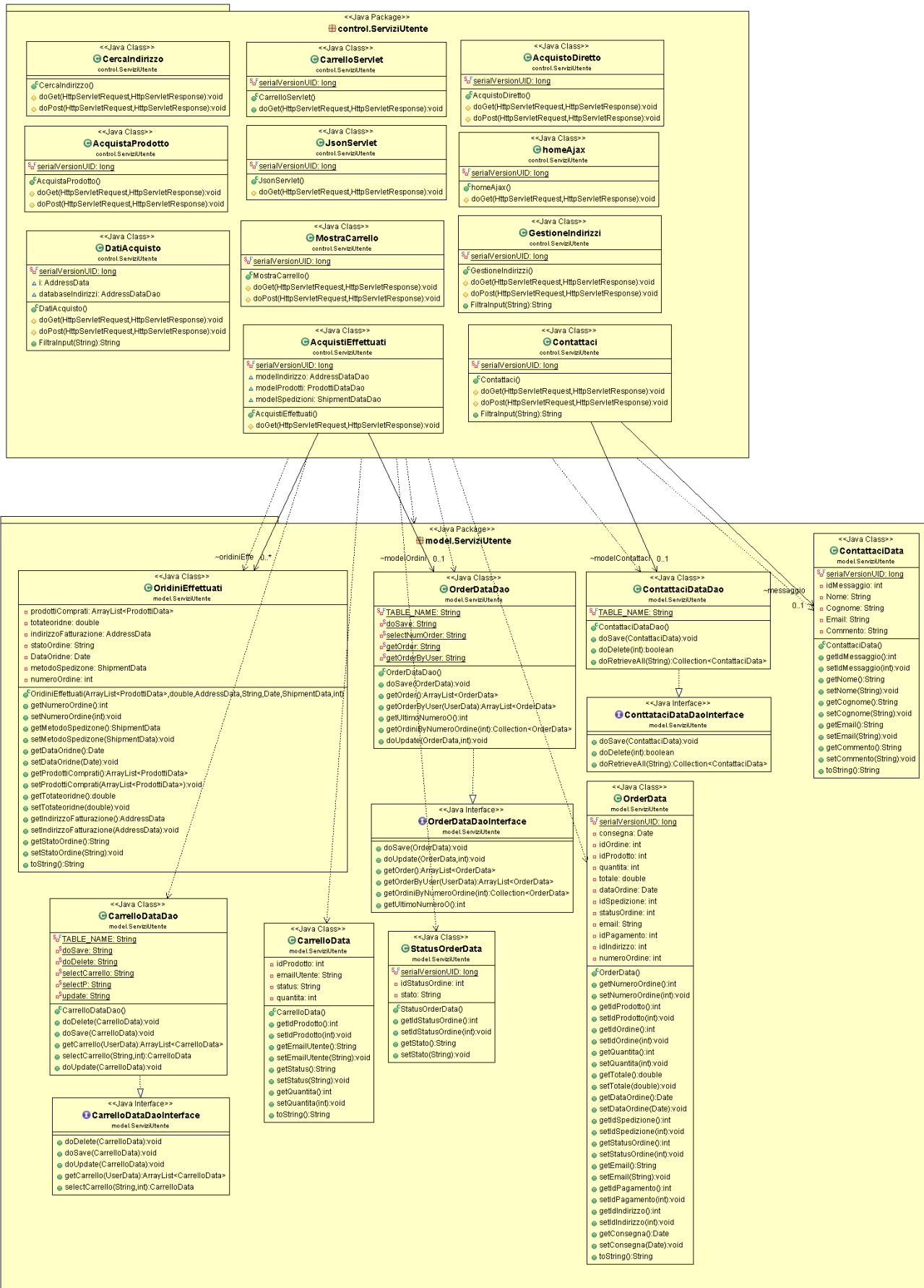
I seguenti diagrammi dei packages sono stati realizzati utilizzando ObjectAID, un plugin del framework di Eclipse, dopo aver completato lo sviluppo.

3.1 Gestione Account



3.2 Gestione Amministratore

3.3 Gestione ServiziUtente



4.0 Glossario

Trade-offs: compromessi;

Package: pacchetto;

Open source: software non protetto da copyright e liberamente modificabile dagli utenti;

ODD: Object Design Document;

Query: interrogazione di un database per estrarre o aggiornare i dati che soddisfano un certo criterio di ricerca .