*<Object Design*

*Document*

*Progetto*

*OctoPlus*



**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 06/12/2023 | **0.1** | **Prima stesura** | **Tutto il team** |

# 1. Introduzione

## 1.1 Object design trade-offs

### 1.1.1 Robustezza vs Tempo

Nei primi tempi non sono previsti picchi di utenza elevati, pertanto si è deciso di ritardare l’implementazione di funzionalità avanzate che renderebbero il sistema più robusto e capace di gestire uno stress maggiore, questo per velocizzare il lancio della prima versione.

### 1.1.2 Sicurezza e Attendibilità vs Tempo

## Si ritiene che la sicurezza dei dati degli utenti registrati e l’attendibilità della piattaforma siano caratteristiche necessarie che la piattaforma dovrebbe avere dalla prima versione al fine di tutelare i clienti. Il tutto, naturalmente, richiede tempo per lo sviluppo il quale sarà sottratto alla robustezza.

## 1.2 Linee guida

Qui di seguito sono riportate alcune linee guide per la stesura del codice:

* .
* .

## 1.3 Referenze

* **R.A.D.**
* **S.D.D.**

2.packeges

# 3. Interfacce di classe

## 3.1 AccountModelDS

|  |  |
| --- | --- |
| AccountModelDS  Cambiare account con utente se va bene | Questa classe permette di interfacciarci al DBMS relazione modificando e interrogando l’entità Utente. |
| Precondizione | **PostCondizione** |
| context AccountModelDS::doRetrieveByKey(Utente utente)  pre: utente.id!= null | context AccountModelDS::doRetrieveByKey(Utente utente)  post: return Utente, se la tupla non esiste nel DB l’oggetto restituito ha tutti i valori settati a null, altrimenti contiene le informazioni ricavate dalla tupla |
| context AccountModelDS::doSave(Account account)  pre: account.email != null and account.email != “”  and account.email è formattata correttamente  and DB not contains account.email  and account.password è formattata correttamente and password conferma== account.password  and account.nome != null and account.nome != “”  and account.cognome != null and account.cognome != “”  and account.numerotelefono è formattato correttamente | context AccountModelDS::doSave(Account account)  post: viene inserita nel DB una tupla contenente le informazioni di account |
| context AccountModelDS::doUpdatePassword (Account account)  pre: passwordvecchiainserita==account.password and  account.passwordnuova != null and account.passwordnuova != “”  and account.passwordnuova è formattata correttamente and account.passwordnuova==passwordconferma | context AccountModelDS::doUpdateInfo(Account account)  post: aggiorna nel DB tutti la tupla corrispondente al campo “password” |