

Java: Sistema di CRM

Gruppo 16:

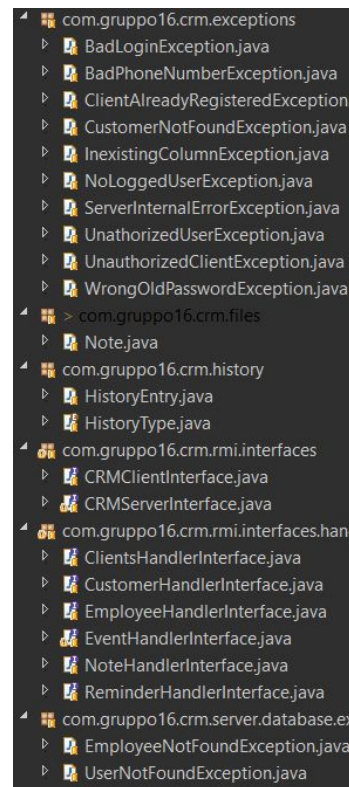
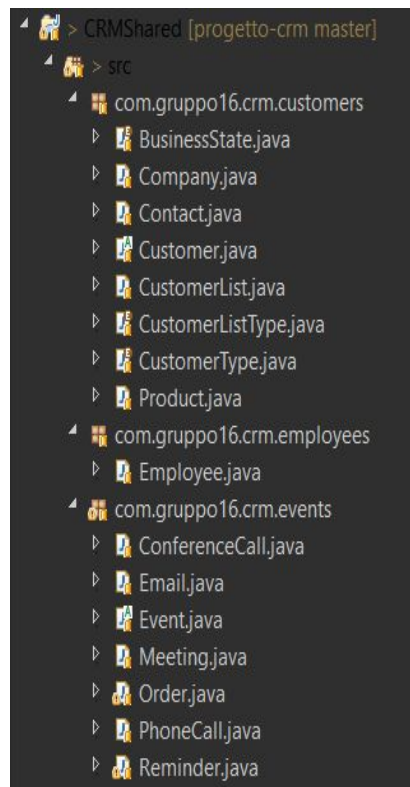
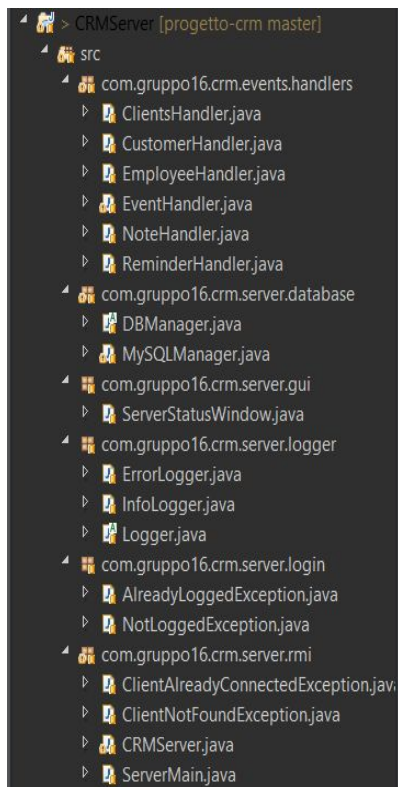
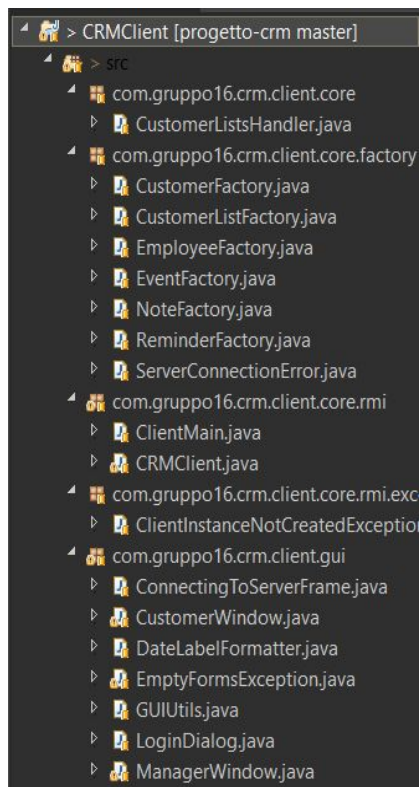
Luca Maltagliati

Marco Turetta

Sergio Cucinotta

Progetti Server Client Shared GUI

- Due parti principali, *CRMServer* e *CRMClient*, realizzati in due progetti Eclipse differenti che includono entrambi il progetto *CRMShared* nel Build Path. *CRMShared* include le interfacce richieste per il funzionamento di RMI e le classi condivise tra Client e Server.



- Nel corso della realizzazione del progetto sono stati sfruttati alcuni *Design Patterns*.

Il Logger è un esempio del pattern *Chain of Responsibility*.

```
public abstract class Logger {

    public static final int INFO = 1;
    public static final int ERROR = 2;

    protected int level;

    protected Logger nextLogger;

    public void setNextLogger(Logger nextLogger) {
        this.nextLogger = nextLogger;
    }

    public void logMessage(int level, HistoryType logType, int employeeID, String message) {
        if (this.level == level) {
            write(logType, employeeID, message);
        }
        if (nextLogger != null) {
            nextLogger.logMessage(level, logType, employeeID, message);
        }
    }

    protected void saveLog(HistoryType logType, int employeeID, String message) {
        try {
            DBManager.getInstance().saveToHistory(logType, employeeID, message);
        } catch (ClassNotFoundException | SQLException e) {
            System.err.println("Impossibile comunicare con il database per salvare il log!");
            e.printStackTrace();
        }
    }

    abstract protected void write(HistoryType logType, int employeeID, String message);
}
```

```
private static Logger getChainOfLoggers() {
    Logger infoLogger = new InfoLogger(Logger.INFO);
    Logger errorLogger = new ErrorLogger(Logger.ERROR);
    infoLogger.setNextLogger(errorLogger);
    return infoLogger;
}
```

```
public class InfoLogger extends Logger {

    public InfoLogger(int level){
        this.level = level;
    }

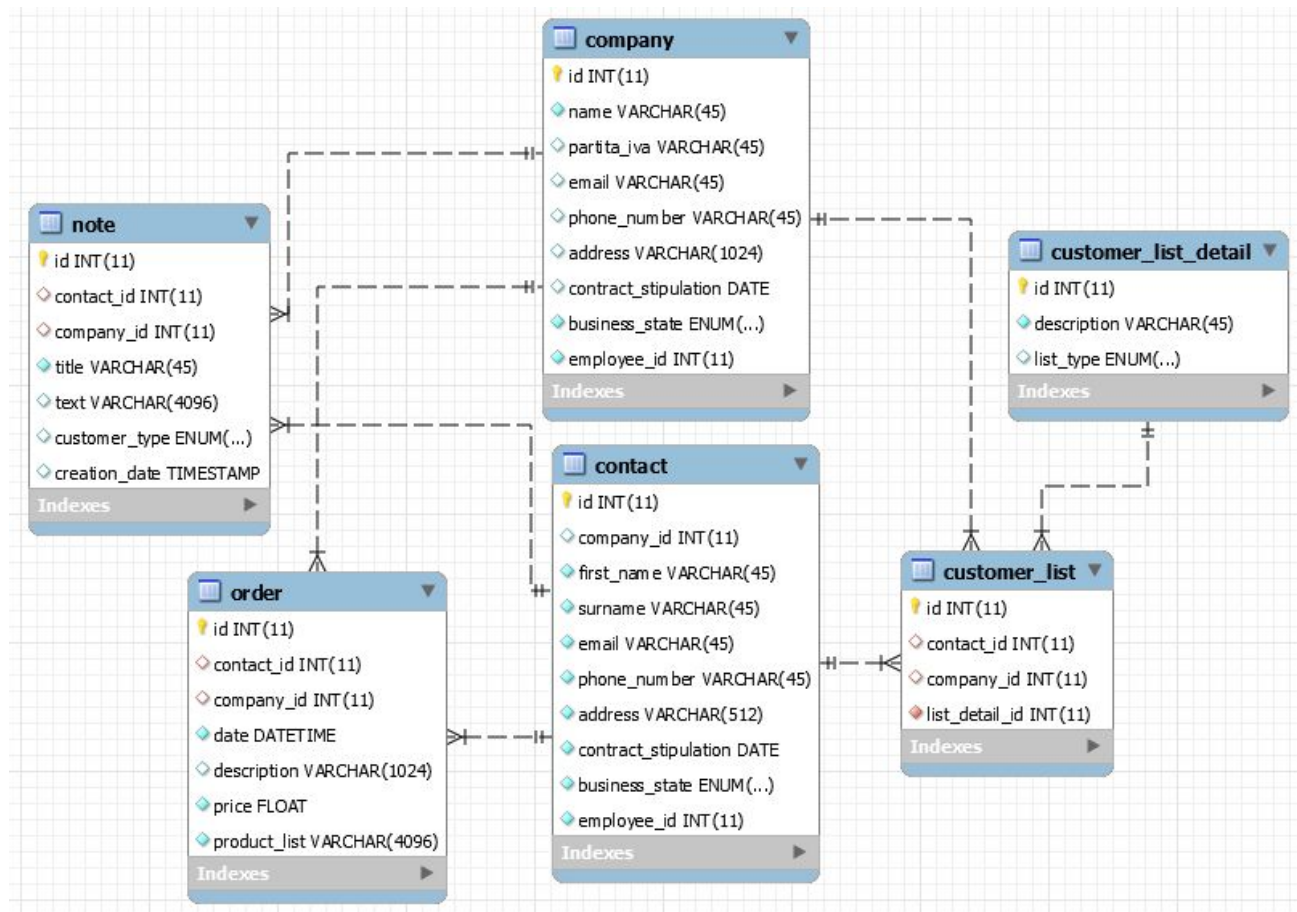
    @Override
    protected void write(HistoryType logType, int employeeID, String message) {
        saveLog(logType, employeeID, message);
        System.out.println("Logger: " + message);
    }
}
```

```
public class ErrorLogger extends Logger {

    public ErrorLogger(int level) {
        this.level = level;
    }

    @Override
    protected void write(HistoryType logType, int employeeID, String message) {
        saveLog(logType, employeeID, message);
        System.err.println("Errore: " + message);
    }
}
```

- Alcune tabelle del database
- Per realizzare il Database abbiamo utilizzato le Foreign Keys.
- Molte di esse hanno l'opzione ON DELETE CASCADE.



- Per la creazione di nuovi oggetti sul Client abbiamo utilizzato il Factory Pattern.

```
13 public class ReminderFactory {
14
15     public ReminderFactory() {
16     }
17
18     public Reminder createReminder(CustomerType type, int customerID, String content, LocalDate expirationDate)
19         throws RemoteException, ServerInternalErrorException, UnauthorizedClientException,
20         UnauthorizedUserException {
21         int reminderID = ClientMain.getClient().saveNewReminder(type, content, expirationDate, customerID);
22         return new Reminder(reminderID, customerID, content, expirationDate);
23     }
24
25 }
```

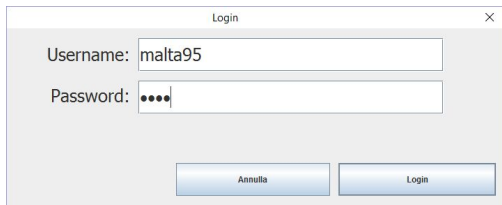

- Interfaccia Client: è utile solo per passare al Server le istanze dei Client.
- Interfaccia Server: è necessaria al Client per poter chiamare i metodi sul server via RMI.
- L'interfaccia Server estende tutte le interfacce degli handler, ereditandone i metodi.

```
10 public interface CRMClientInterface extends Remote {  
11  
12 }
```

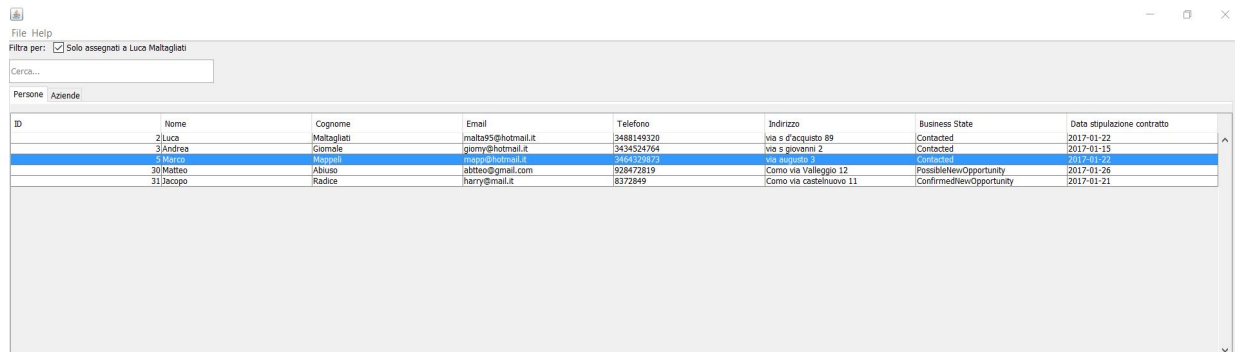
```
12 public interface CRMServerInterface extends Remote, EmployeeHandlerInterface, EventHandlerInterface,  
13     NoteHandlerInterface, ReminderHandlerInterface, CustomerHandlerInterface, ClientsHandlerInterface {  
14 }
```

```
12 public interface NoteHandlerInterface {  
13  
14     public void deleteNote(CRMClientInterface client, Note note)  
15         throws RemoteException, ServerInternalErrorException, UnauthorizedClientException, UnauthorizedUserException;  
16  
17     int saveNewNote(CustomerType type, CRMClientInterface client, String title, String content, int customerID)  
18         throws RemoteException, ServerInternalErrorException, UnauthorizedClientException, UnauthorizedUserException;  
19  
20     public void updateNoteValue(CRMClientInterface client, int noteID, String toUpdate, Object newValue)  
21         throws RemoteException, ServerInternalErrorException, UnauthorizedClientException, UnauthorizedUserException;  
22  
23 }
```

- Per realizzare l'interfaccia grafica abbiamo utilizzato la libreria Swing, aiutandoci con il plug-in di Eclipse Window Builder

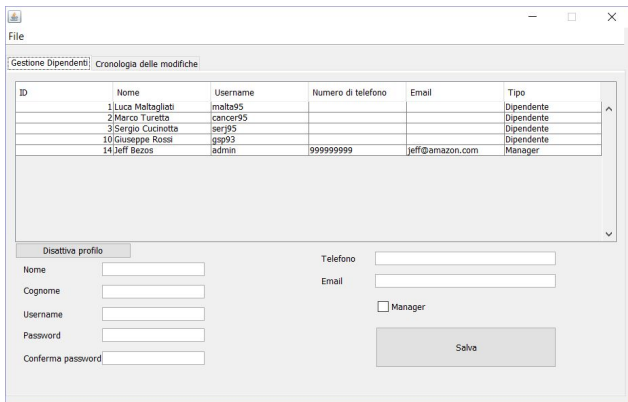


A login dialog box titled "Login" with a close button (X). It contains two text input fields: "Username:" with the value "malta95" and "Password:" with masked characters "••••". Below the fields are two buttons: "Annulla" (Cancel) and "Login".



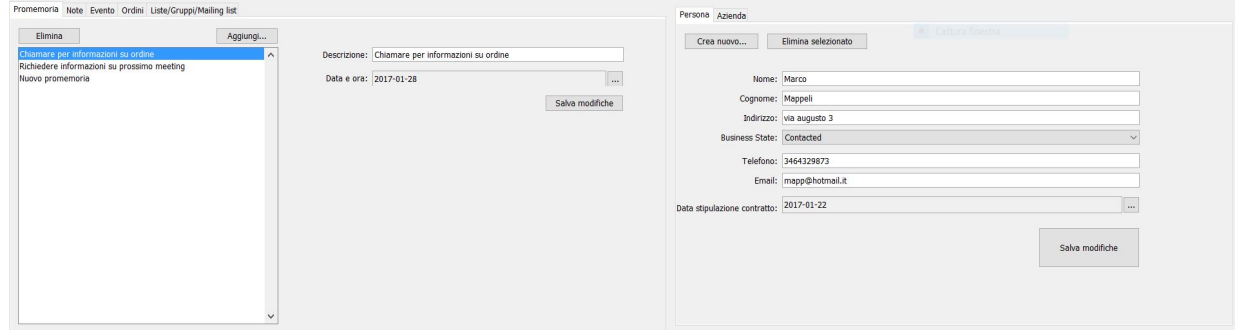
The main application window has a menu bar (File, Help) and a filter section: "Filtra per: ☒ Solo assegnati a Luca Maltagliati". Below is a search bar "Cerca...". A tabbed interface shows "Persone" and "Aziende". The "Persone" tab displays a table with columns: ID, Nome, Cognome, Email, Telefono, Indirizzo, Business State, and Data stipulazione contratto.

ID	Nome	Cognome	Email	Telefono	Indirizzo	Business State	Data stipulazione contratto
2	Luca	Maltagliati	malta95@hotmail.it	3488149320	via s d'acquisto 89	Contacted	2017-01-22
3	Andrea	Giromale	giromy@hotmail.it	3434524764	via s giovanni 2	Contacted	2017-01-15
4	Luca	Giromale	luca@giromale.it	0155326123	via s d'acquisto 89	Contacted	2017-01-22
30	Matteo	Abbaso	matteo@gmail.com	028472819	Corno via Valleggio 12	PossibleNewOpportunity	2017-01-26
31	Jacopo	Radice	harry@mail.it	8372849	Corno via castelnovo 11	ConfirmedNewOpportunity	2017-01-21



A window titled "Gestione Dipendenti" with a subtitle "Cronologia delle modifiche". It contains a table with columns: ID, Nome, Username, Numero di telefono, Email, and Tipo. Below the table is a "Disattiva profilo" button. To the right are input fields for "Nome", "Cognome", "Username", "Password", and "Conferma password", along with "Telefono" and "Email" fields, a "Manager" checkbox, and a "Salva" button.

ID	Nome	Username	Numero di telefono	Email	Tipo
1	Luca Maltagliati	malta95			Dipendente
2	Marco Turetta	canon95			Dipendente
3	Sergio Cucinotta	ser95			Dipendente
10	Giuseppe Rossi	gsp93			Dipendente
14	Jeff Bezos	admin	999999999	jeff@amazon.com	Manager



The bottom section contains two windows. The left window, "Promemoria", has tabs for "Note", "Evento", "Ordini", "Liste/Gruppi/Mailing list", and "Aggiungi...". It shows a description "Chiamare per informazioni su ordine" and a date "2017-01-28". The right window, "Persona", has tabs for "Persona" and "Azienda". It shows details for "Marco Mappelli" with fields for "Nome", "Cognome", "Indirizzo", "Business State", "Telefono", "Email", and "Data stipulazione contratto".