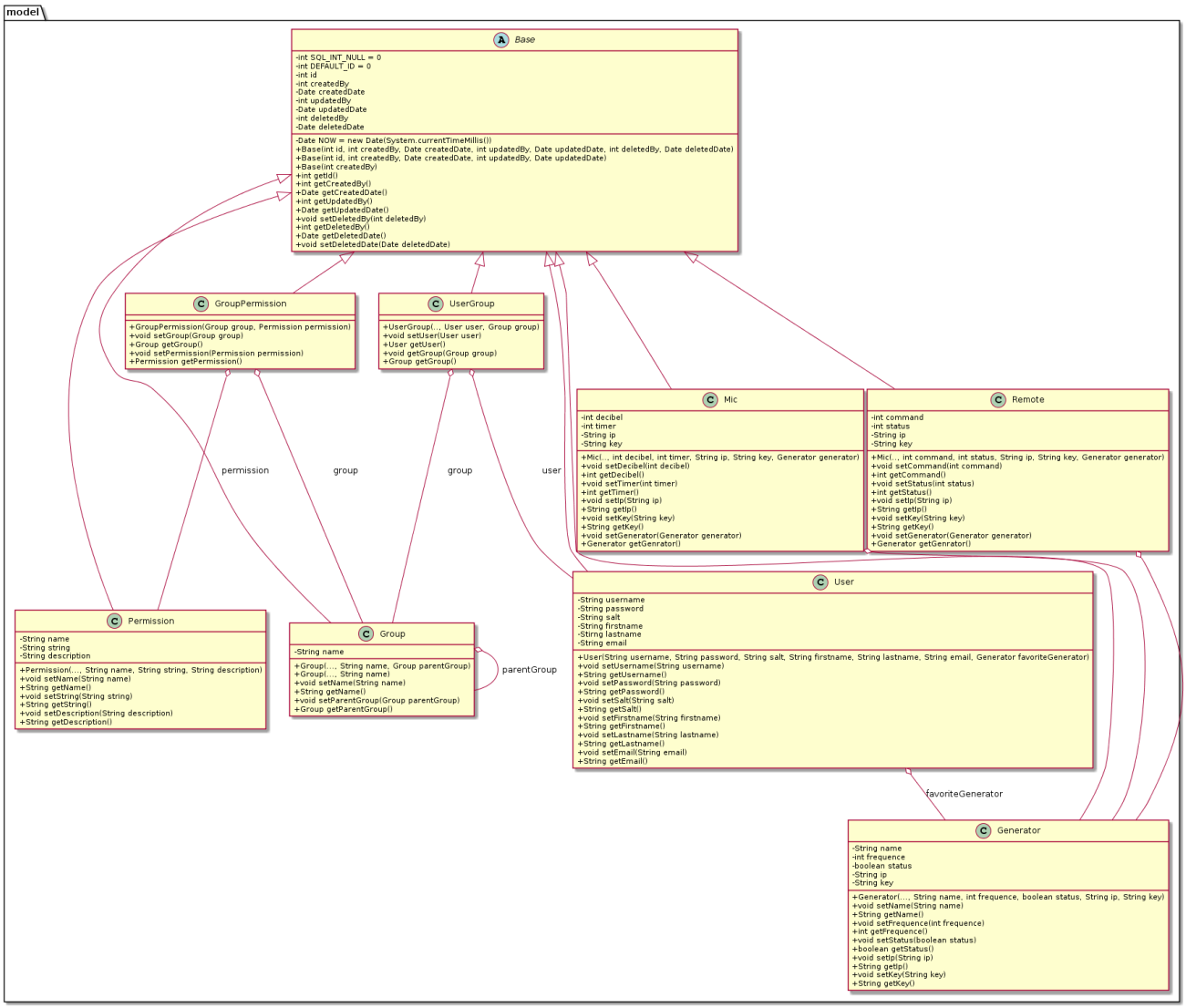
Diario di lavoro

Luogo: Canobbio

Data: 08.10.2019

Lavori svolti

Ho finito di implementare le classi dei modelli dei dati. Ed ho aggiornato il diagramma delle classi:



Dopo di che ho iniziato a sviluppare le classi di astrazione della connessione database.

DaoException

Ho implementato la classe `DaoException`, che estende semplicemente la classe`Exception`, verrà utilizzata per le eccezioni generali del modello dao.

DbDao

Poi ho iniziato a scrivere la classe `DbDao` che sarà la classe principale di comunicazione con il database. L'idea è che la maggior parte delle operazioni da eseguire per inserire, aggiornare e selezionare i dati nel

database, vengano scritte in una sola classe, e non debbano venir riscritte per ogni classe.
Cosi facendo si scrive meno codice. Ma la classe di base è molto complicata.

Per il momento ho creato la classe con alcuni metodi di aiuto che serviranno poi in futuro:

```
protected Date getDate(ResultSet resultSet, String column) throws
SQLException {
    if (resultSet != null) {
        Timestamp timestamp = resultSet.getTimestamp(column);
        if (timestamp != null) {
        return new Date(resultSet.getTimestamp(column).getTime());
        } else {
            return null;
        }
    } else {
        throw new SQLException("Connection closed");
    }
}
```

Metodo che serve per richiedere una data da un a colonna di un sql result set.

```
protected Timestamp getTimestamp(Date date) {
    if (date != null) {
        return new Timestamp(date.getTime());
    } else {
        return null;
    }
}
```

Trasforma un oggetto data di java in oggetto Timestamp di SQL.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Sono ancora avanti rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Finire di scrivere la classe di `DbDao`