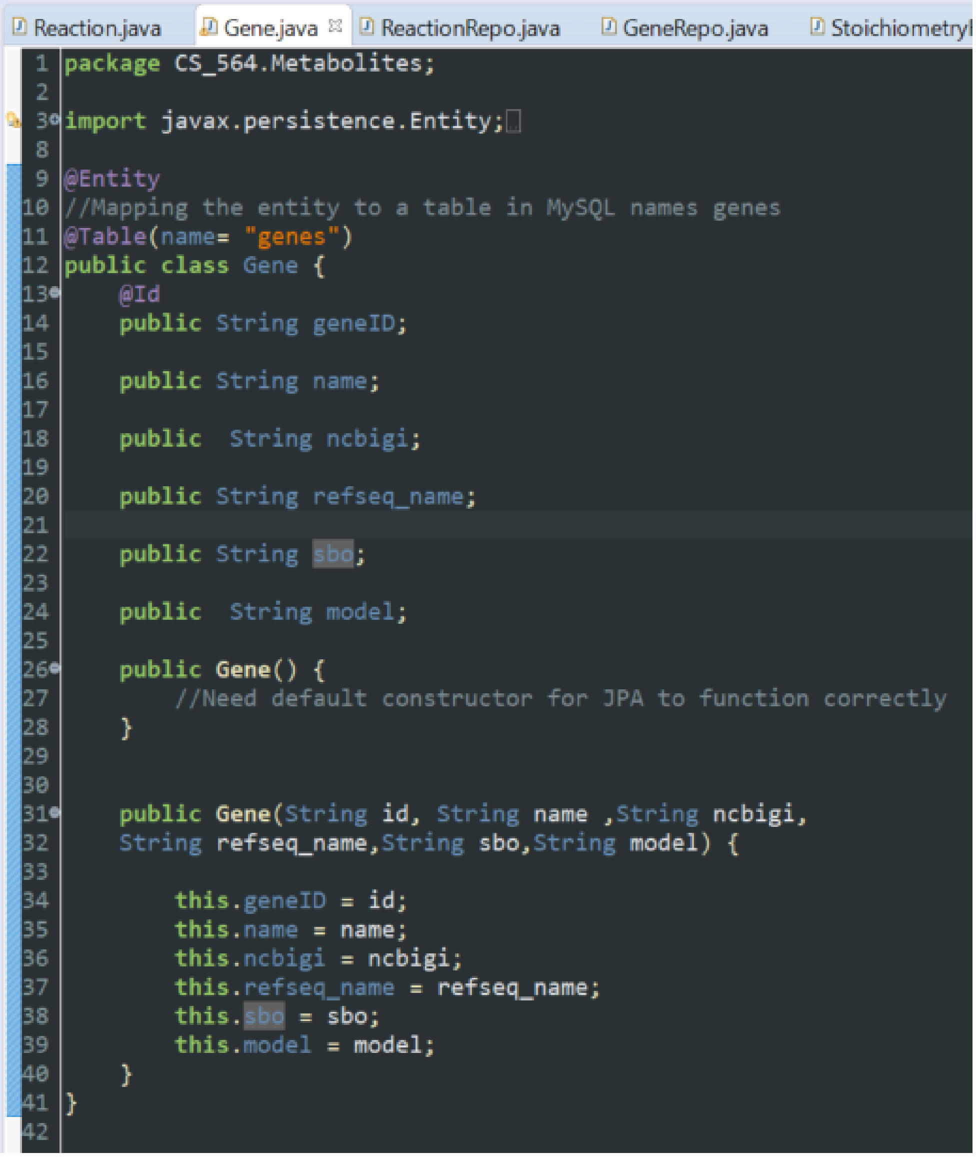
Database Creation

We created our database using the Java Persistence API and Hibernate. With this you can write Java classes that map to tables in MySQL and there’s a setting you can choose to make it create a table in MySQL with the name of the table you told the Java class to map to if that table does not already exist in the database. So this is how we created the tables in our database. For example the Gene class below maps to the genes table in our database. When we created this class the genes table did not exist yet in our database. Then when parsing the JSON files from our data source, we used the built in JPA save() to save each gene object to the database. The first time save() was used the gene table was created in our database. The attributes in the gene table have the same names as the field in the Gene Object. We did this for all our tables in our database. The code for paring the JSON and populating the database is in the Cs564Application.java file.



```java
package CS_564.Metabolites;

import javax.persistence.Entity;

@Entity
//Mapping the entity to a table in MySQL names genes
@Table(name= "genes")
public class Gene {
    @Id
    public String geneID;

    public String name;

    public  String ncbigi;

    public String refseq_name;

    public String sbo;

    public  String model;

    public Gene() {
        //Need default constructor for JPA to function correctly
    }


    public Gene(String id, String name ,String ncbigi,
    String refseq_name,String sbo,String model) {

        this.geneID = id;
        this.name = name;
        this.ncbigi = ncbigi;
        this.refseq_name = refseq_name;
        this.sbo = sbo;
        this.model = model;
    }
}
```

*We forgot to add the foreign keys when we initially create ur database, so we had to add them in with these statements

```
#Six SQL Statements to add the foreign key relationships

alter table stoichiometry add foreign key (reactionid) references reactions(reactionid) on delete cascade on update cascade;

alter table has add foreign key (reactionid) references reactions(reactionid) on delete cascade on update cascade;

alter table metabolites add foreign key (bigg_compoundid) references compounds(biggmetaboliteid) on delete set null on update cascade;

alter table linked add foreign key (metaboliteid) references metabolites(metaboliteid) on delete cascade on update cascade;

alter table end_meta_reaction add foreign key (metaboliteid) references metabolites(metaboliteid) on delete cascade on update cascade, add foreign key (reactionid) references reactions(reactionid) on delete cascade on update cascade;

alter table start_meta_reaction add foreign key (metaboliteid) references metabolites(metaboliteid) on delete cascade on update cascade, add foreign key (reactionid) references reactions(reactionid) on delete cascade on update cascade;
```