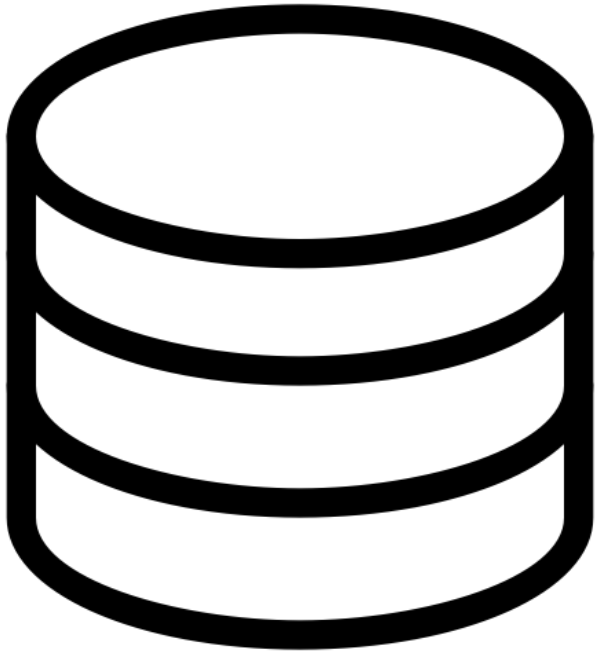
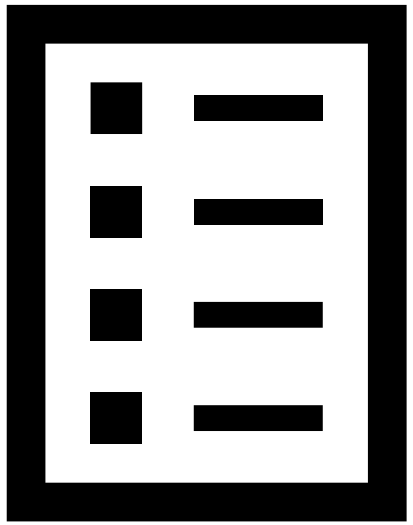
Zwischenpräsentation

Gliederung

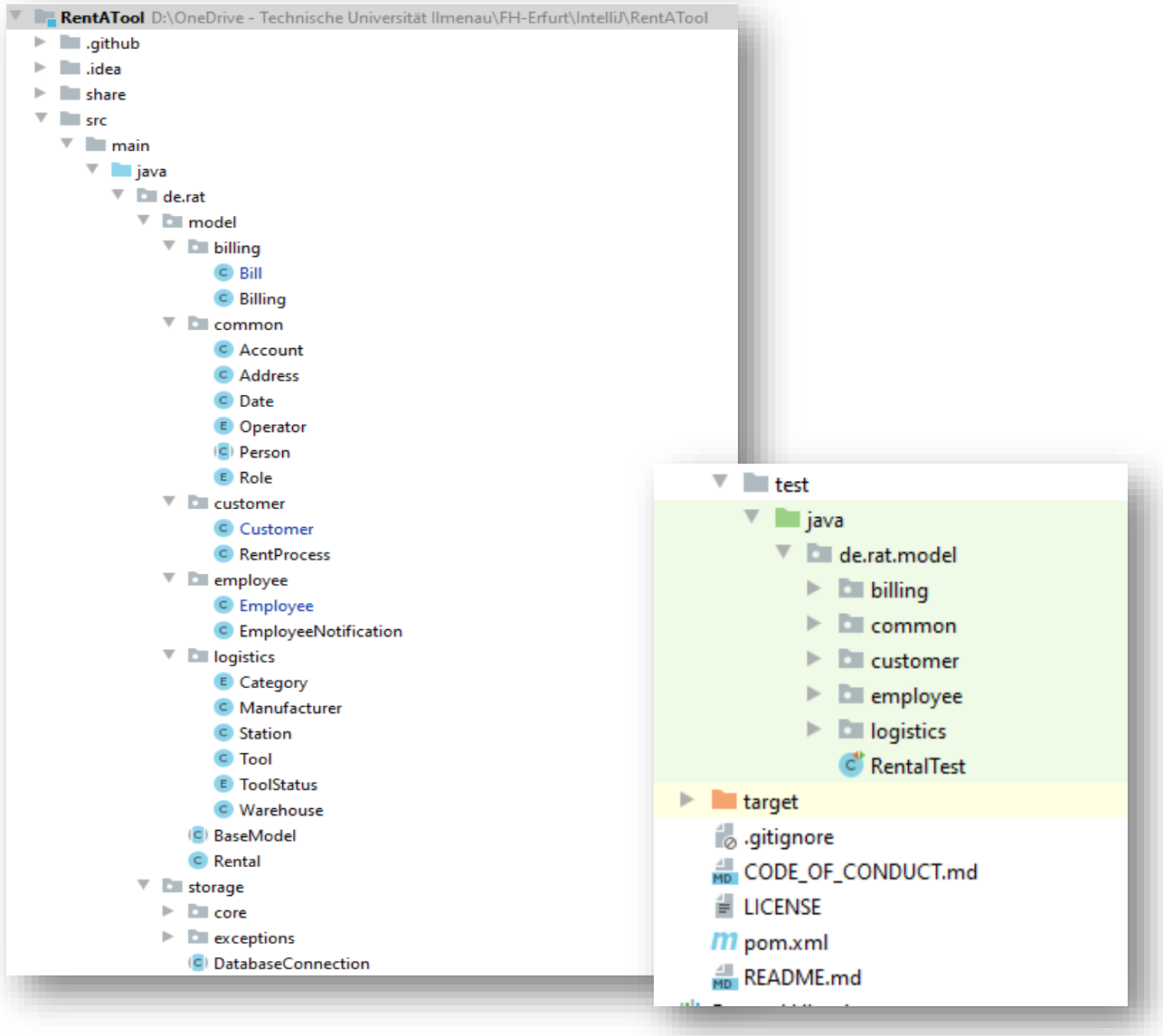
Projektteam & Kundenunternehmen

Projektstruktur & Datenbankmodel

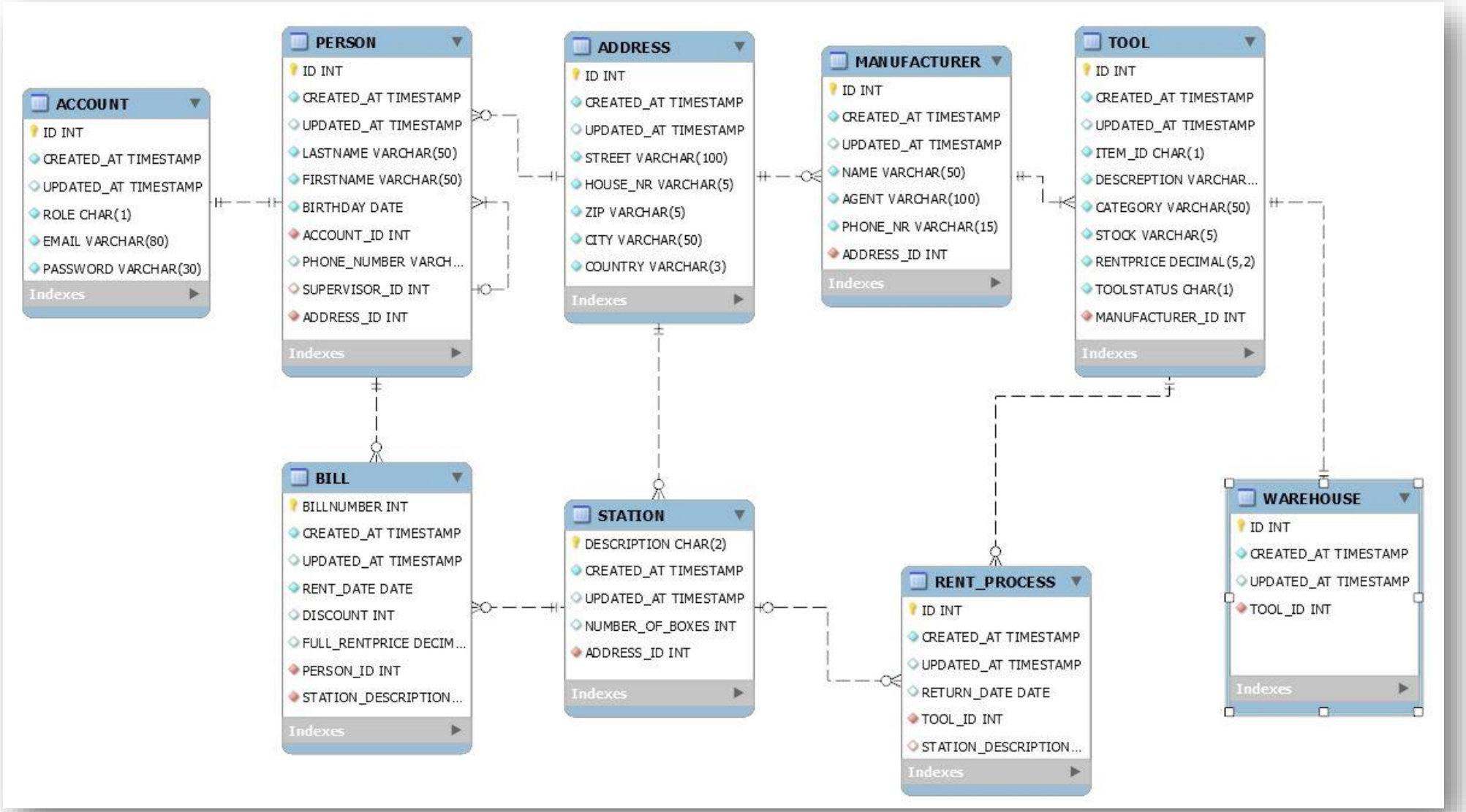




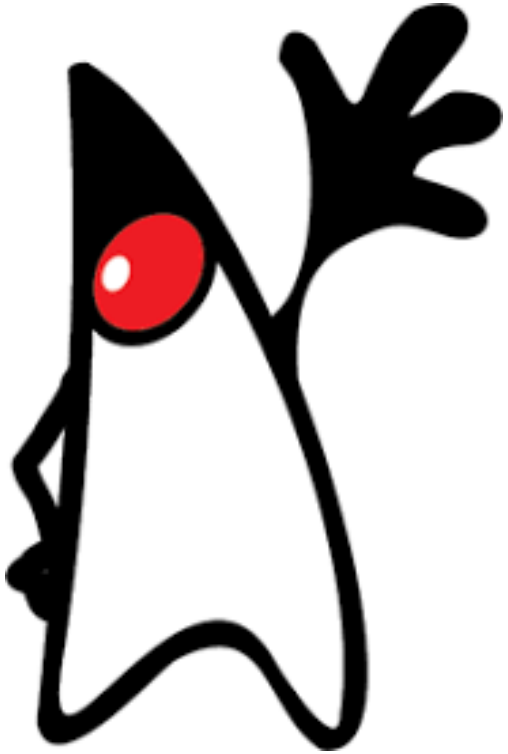
Projektstruktur & Datenbankmodel



Projektstruktur & Datenbankmodel



Code & GitHub



Code & GitHub


```java
public class DataSource{
    private static final String PERSISTENCE_UNIT_NAME = "rat-pu";
    private EntityManagerFactory entityManagerFactory;
    private static DataSource dataSourceUniqueInstance;

    private DataSource(){
        this.entityManagerFactory = Persistence.createEntityManagerFactory( PERSISTENCE_UNIT_NAME );
        entityManagerFactory.createEntityManager();
    }
    public static DataSource getDataSource(){
        if(dataSourceUniqueInstance ==null){
            dataSourceUniqueInstance =new DataSource();
        }
        return dataSourceUniqueInstance;
    }
    public EntityManager getEntityManager() { return  entityManagerFactory.createEntityManager(); }

}
```

Code & GitHub


```java
public abstract class Repository<T extends BaseModel> {

    private DataSource dataSource;

    public Repository() { dataSource = DataSource.getDataSource(); }

    //String can be also entire object we want to change
    @PreUpdate
    protected abstract void updateOperation(T model, String argument);


    public Integer create(T model) {
        EntityManager entityManager = dataSource.getEntityManager();
        entityManager.getTransaction().begin();
        entityManager.persist(model);
        entityManager.flush();
        entityManager.getTransaction().commit();
        return model.getId();

    }
    public void update(T model, String argument) {
        EntityManager entityManager = dataSource.getEntityManager();
        entityManager.getTransaction().begin();
        updateOperation(model, argument);
        entityManager.merge(model);
        entityManager.getTransaction().commit();
    }
```

```java
public void delete(T model) {
    EntityManager entityManager = dataSource.getEntityManager();
    entityManager.getTransaction().begin();
    model=entityManager.merge(model);


    entityManager.remove(model);
    entityManager.getTransaction().commit();
}
```

```java
public T findOne(T model){
    EntityManager entityManager = dataSource.getEntityManager();
    return (T) entityManager.find(model.getClass(), model.getId());
}


public List<T> getAll(String className){
    EntityManager entityManager = dataSource.getEntityManager();
    Query query = entityManager.createQuery(
            s: "SELECT c FROM "+className+" c");
    return (List<T>) query.getResultList();
}

public void delete( List<T> entries )
{
    EntityManager entityManager = dataSource.getEntityManager();
    entityManager.getTransaction().begin();
    for( T entry : entries )
    {
        entityManager.remove( entry );
    }
    entityManager.getTransaction().commit();
}
```

Code & GitHub


```java
public abstract class ToolRepository extends Repository<Tool> {
    @Override
    protected void updateOperation(Tool model, String description) {
        model.setDescription(description);

    }
```

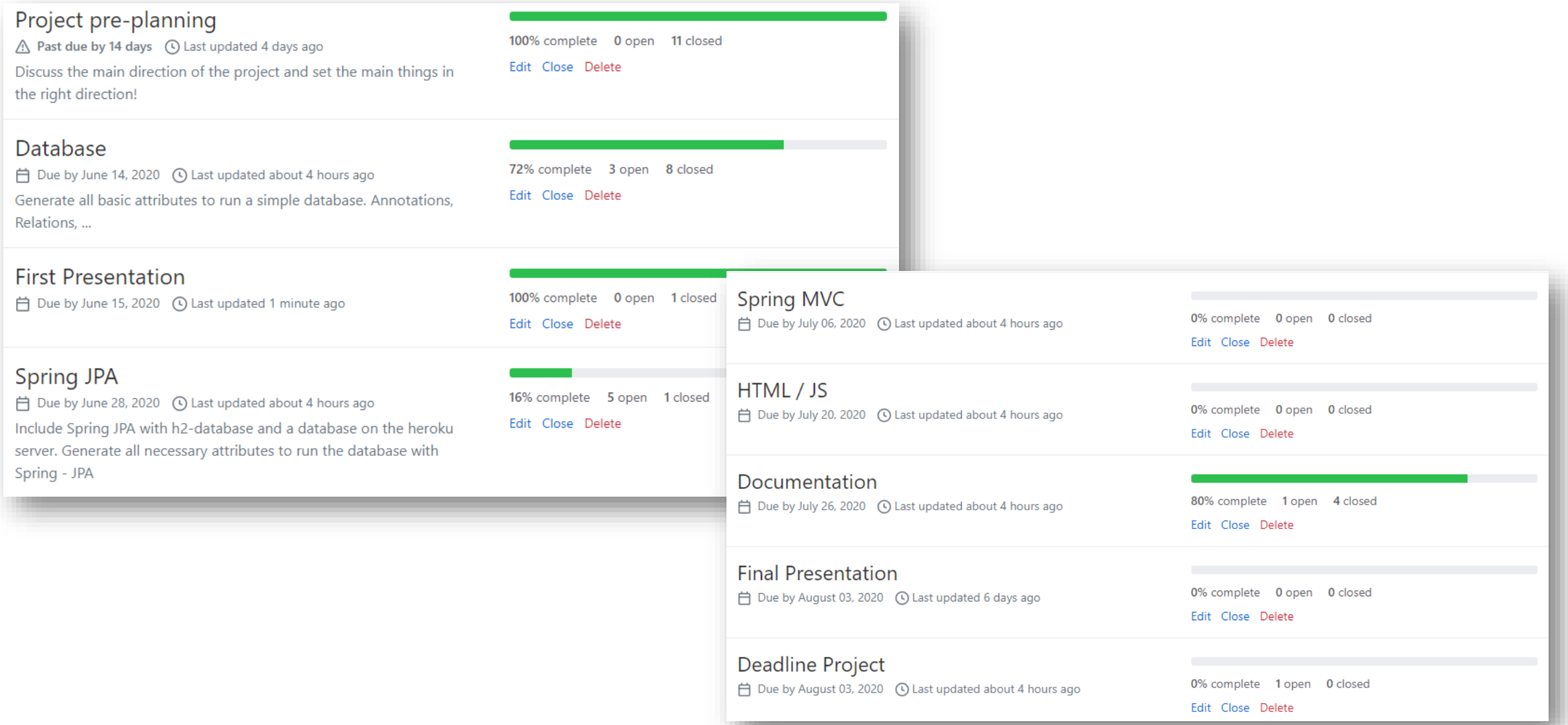
Code & GitHub


```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
             version="2.0">

    <persistence-unit name="rat-pu" transaction-type="RESOURCE_LOCAL">

        <description>Hibernate EntityManager</description>
        <exclude-unlisted-classes>false</exclude-unlisted-classes>
        <properties>
            <property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>
            <property name="hibernate.hbm2ddl.auto" value="update"/>
            <property name="javax.persistence.jdbc.driver" value="org.h2.Driver"/>
            <property name="javax.persistence.jdbc.url" value="jdbc:h2:mem:addressbook;DB_CLOSE_DELAY=-1"/>
            <property name="javax.persistence.jdbc.user" value="sa"/>
        </properties>
    </persistence-unit>

</persistence>
```

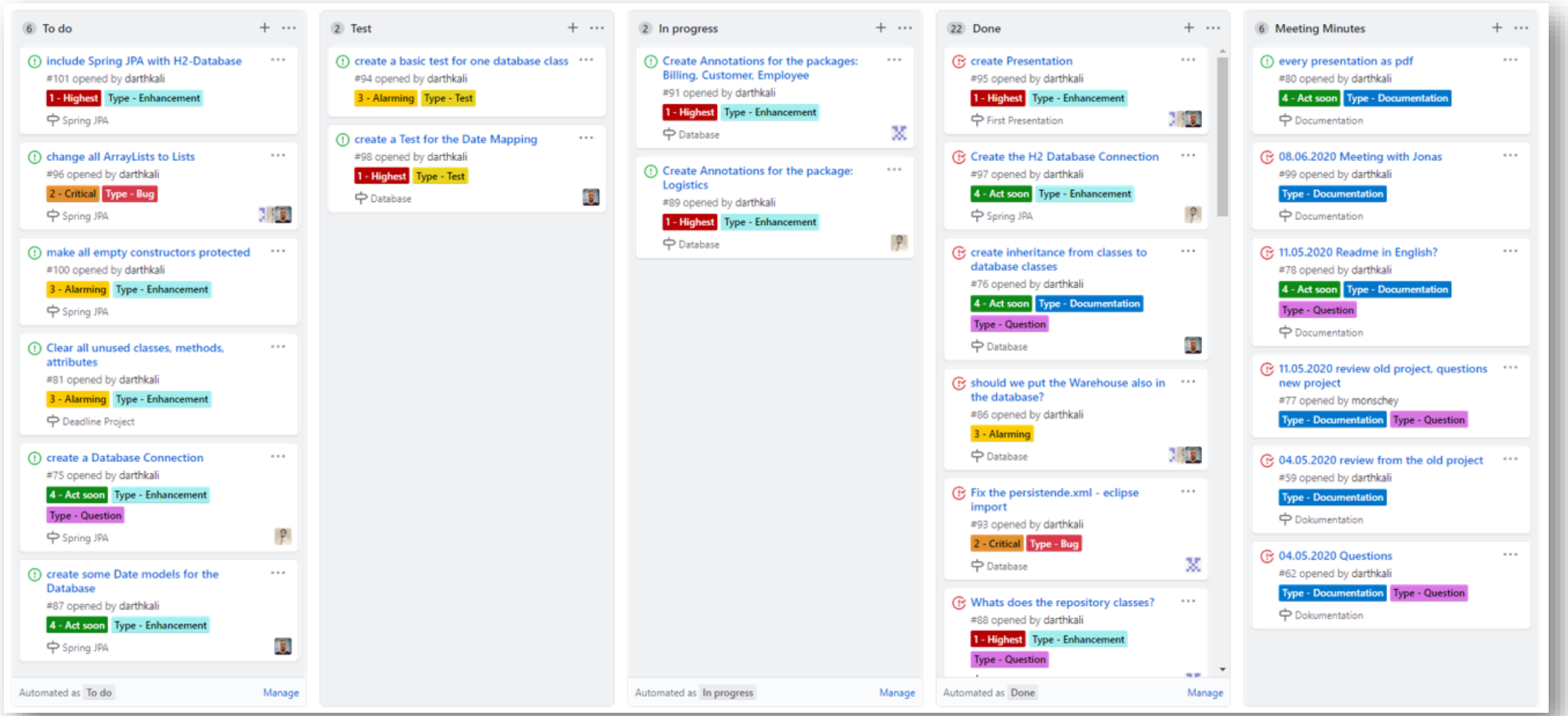
Code & GitHub


```java
public abstract class BaseModel {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Temporal(TemporalType.TIMESTAMP)
    private Date created;

    @Temporal(TemporalType.TIMESTAMP)
    private Date modified;

    @PrePersist
    void onCreate() { this.setCreated(new Date()); }

    @PreUpdate
    void onUpdate() { this.setModified(new Date()); }

    public void setCreated(Date created) {this.created = created;}
    public void setModified(Date modified) {this.modified = modified;}

    public int getId() {return id;}
    public Date getCreated() {return created;}
    public Date getModified() {return modified;}
}
```

Code & GitHub


Project pre-planning
Past due by 14 days
Last updated 4 days ago
Discuss the main direction of the project and set the main things in the right direction!
100% complete 0 open 11 closed
Edit Close Delete
Database
Due by June 14, 2020
Last updated about 4 hours ago
Generate all basic attributes to run a simple database. Annotations, Relations, ...
72% complete 3 open 8 closed
Edit Close Delete
First Presentation
Due by June 15, 2020
Last updated 1 minute ago
100% complete 0 open 1 closed
Edit Close Delete
Spring JPA
Due by June 28, 2020
Last updated about 4 hours ago
Include Spring JPA with h2-database and a database on the heroku server. Generate all necessary attributes to run the database with Spring - JPA
16% complete 5 open 1 closed
Edit Close Delete
Spring MVC
Due by July 06, 2020
Last updated about 4 hours ago
0% complete 0 open 0 closed
Edit Close Delete
HTML / JS
Due by July 20, 2020
Last updated about 4 hours ago
0% complete 0 open 0 closed
Edit Close Delete
Documentation
Due by July 26, 2020
Last updated about 4 hours ago
80% complete 1 open 4 closed
Edit Close Delete
Final Presentation
Due by August 03, 2020
Last updated 6 days ago
0% complete 0 open 0 closed
Edit Close Delete
Deadline Project
Due by August 03, 2020
Last updated about 4 hours ago
0% complete 1 open 0 closed
Edit Close Delete

Code & GitHub


6 To do
include Spring JPA with H2-Database
#101 opened by darthkali
1 - Highest Type - Enhancement
Spring JPA
change all ArrayLists to Lists
#96 opened by darthkali
2 - Critical Type - Bug
Spring JPA
make all empty constructors protected
#100 opened by darthkali
3 - Alarming Type - Enhancement
Spring JPA
Clear all unused classes, methods, attributes
#81 opened by darthkali
3 - Alarming Type - Enhancement
Deadline Project
create a Database Connection
#75 opened by darthkali
4 - Act soon Type - Enhancement
Type - Question
Spring JPA
create some Date models for the Database
#87 opened by darthkali
4 - Act soon Type - Enhancement
Spring JPA
Automated as To do
Manage
2 Test
create a basic test for one database class
#94 opened by darthkali
3 - Alarming Type - Test
create a Test for the Date Mapping
#98 opened by darthkali
1 - Highest Type - Test
Database
2 In progress
Create Annotations for the packages: Billing, Customer, Employee
#91 opened by darthkali
1 - Highest Type - Enhancement
Database
Create Annotations for the package: Logistics
#89 opened by darthkali
1 - Highest Type - Enhancement
Database
Automated as In progress
Manage
22 Done
create Presentation
#95 opened by darthkali
1 - Highest Type - Enhancement
First Presentation
Create the H2 Database Connection
#97 opened by darthkali
4 - Act soon Type - Enhancement
Spring JPA
create inheritance from classes to database classes
#76 opened by darthkali
4 - Act soon Type - Documentation
Type - Question
Database
should we put the Warehouse also in the database?
#86 opened by darthkali
3 - Alarming
Database
Fix the persistende.xml - eclipse import
#93 opened by darthkali
2 - Critical Type - Bug
Database
Whats does the repository classes?
#88 opened by darthkali
1 - Highest Type - Enhancement
Type - Question
Automated as Done
Manage
6 Meeting Minutes
every presentation as pdf
#80 opened by darthkali
4 - Act soon Type - Documentation
Documentation
08.06.2020 Meeting with Jonas
#99 opened by darthkali
Type - Documentation
Documentation
11.05.2020 Readme in English?
#78 opened by darthkali
4 - Act soon Type - Documentation
Type - Question
Documentation
11.05.2020 review old project, questions new project
#77 opened by monschey
Type - Documentation Type - Question
04.05.2020 review from the old project
#59 opened by darthkali
Type - Documentation
Dokumentation
04.05.2020 Questions
#62 opened by darthkali
Type - Documentation Type - Question
Dokumentation

Ausblick

Geplanter Ablauf

Spring MVC

Spring Boot

HIBERNATE

REST API

JAX-RS

Einführung Spring → REST Webservices → Webservice oder Website → ???

Vielen Dank