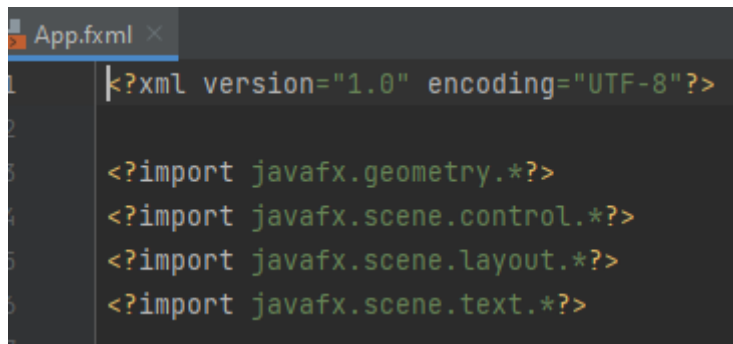
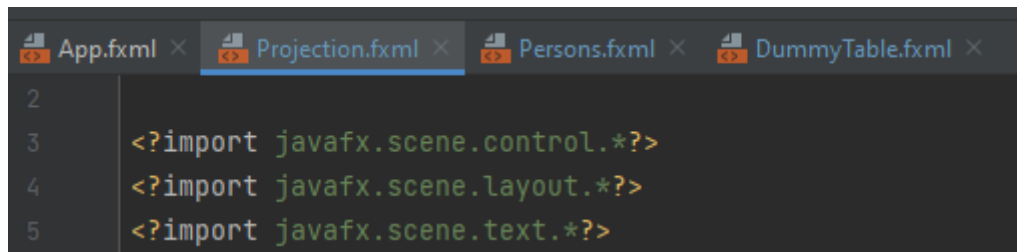


# Cinema app

- You have to implement a Desktop application in Java or Python. For Java use Swing (taught in BPC-PC2T) or JavaFX, for Python, it is up to you. If you already know some web-based frameworks Spring/Jakarta EE/Django/Flask and arbitrary JavaScript library/framework you can also use them.

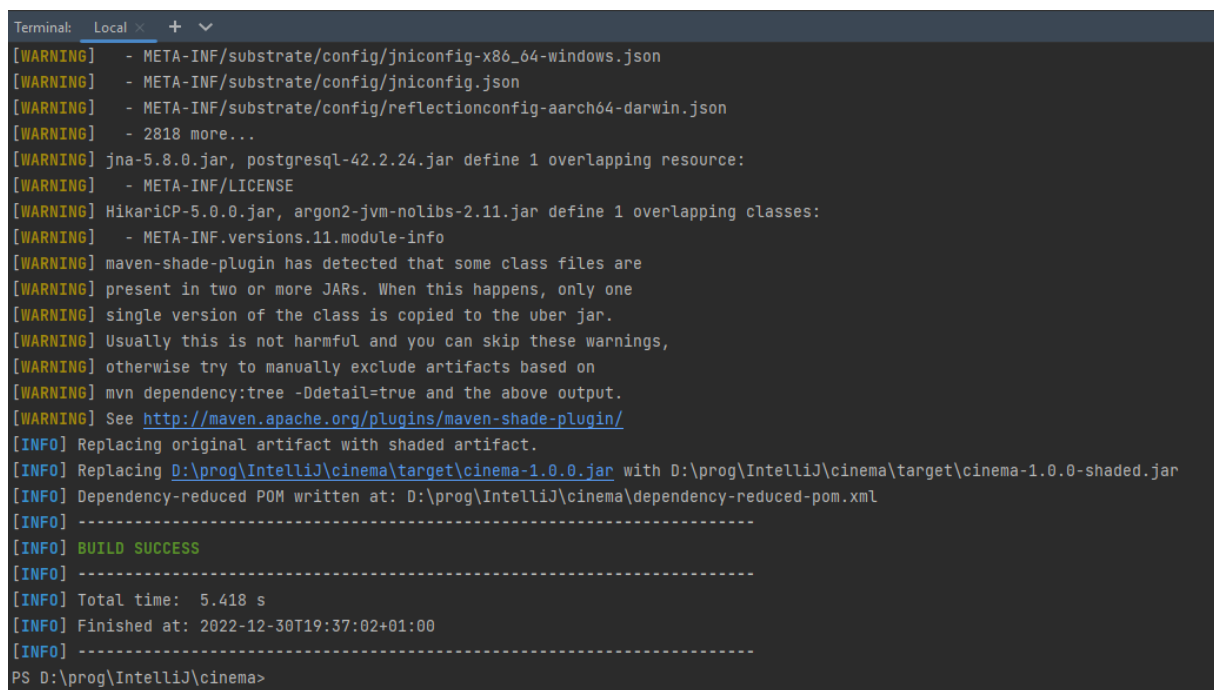


```
App.fxml x
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.*?>
4  <?import javafx.scene.control.*?>
5  <?import javafx.scene.layout.*?>
6  <?import javafx.scene.text.*?>
```



```
App.fxml x Projection.fxml x Persons.fxml x DummyTable.fxml x
2
3  <?import javafx.scene.control.*?>
4  <?import javafx.scene.layout.*?>
5  <?import javafx.scene.text.*?>
```

- The application must be compilable and runnable from command-line (use Maven or Gradle – check seminar project), e.g., to compile/build `mvn clean package` and to run `java -jar my-bds-app.jar`.



```
Terminal: Local x + v
[WARNING] - META-INF/substrate/config/jniconfig-x86_64-windows.json
[WARNING] - META-INF/substrate/config/jniconfig.json
[WARNING] - META-INF/substrate/config/reflectionconfig-aarch64-darwin.json
[WARNING] - 2818 more...
[WARNING] jna-5.8.0.jar, postgresql-42.2.24.jar define 1 overlapping resource:
[WARNING] - META-INF/LICENSE
[WARNING] HikariCP-5.0.0.jar, argon2-jvm-nolib-2.11.jar define 1 overlapping classes:
[WARNING] - META-INF/versions.11/module-info
[WARNING] maven-shade-plugin has detected that some class files are
[WARNING] present in two or more JARs. When this happens, only one
[WARNING] single version of the class is copied to the uber jar.
[WARNING] Usually this is not harmful and you can skip these warnings,
[WARNING] otherwise try to manually exclude artifacts based on
[WARNING] mvn dependency:tree -Ddetail=true and the above output.
[WARNING] See http://maven.apache.org/plugins/maven-shade-plugin/
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing D:\prog\IntelliJ\cinema\target\cinema-1.0.0.jar with D:\prog\IntelliJ\cinema\target\cinema-1.0.0-shaded.jar
[INFO] Dependency-reduced POM written at: D:\prog\IntelliJ\cinema\dependency-reduced-pom.xml
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.418 s
[INFO] Finished at: 2022-12-30T19:37:02+01:00
[INFO] -----
PS D:\prog\IntelliJ\cinema>
```

```

PS D:\prog\IntelliJ\cinema> java -jar target/cinema-1.0.0.jar
HikariPool-1 - configuration:
allowPoolSuspension.....false
autoCommit.....true
catalog.....none
connectionInitSql.....none
connectionTestQuery.....none
connectionTimeout.....30000
dataSource.....none
dataSourceClassName.....none
dataSourceJNDI.....none
dataSourceProperties.....{password=<masked>}
driverClassName.....none
exceptionOverrideClassName.....none
healthCheckProperties.....{}
healthCheckRegistry.....none
idleTimeout.....600000
initializationFailTimeout.....1
isolateInternalQueries.....false
jdbcUrl.....jdbc:postgresql://localhost:5432/dbs-projekt
keepaliveTime.....0
leakDetectionThreshold.....0

```

- The database will contain the user passwords in a hash form (hashed using Argon2, PBKDF2 or any other recommended hashing algorithm).

user_name	user_password
character varying (30)	character varying (256)
tdecourtney0	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
kfarilamb1	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
mfairman2	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
aslott3	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
hvesque4	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
hfitzpayn5	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
cdameisele6	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
dpicheford7	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
cblagini8	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
rbrookz9	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
sclementuccia	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
cmacalpyneb	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
tzanazzic	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
bbarded	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
mschulze	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
kswanwickf	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
ibrettleg	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
jtonksh	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
hkollatschi	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
sscandrootj	\$argon2id\$v=19\$m=65536,t=22,p=1\$IA8LCjpbq/I9IntiHb9Nydg\$6j4/X6oD...
manager	\$argon2id\$v=19\$m=65536,t=22,p=1\$Ku9dzlz7xSD90sCUIKsYSw\$ePJtl1j...

```

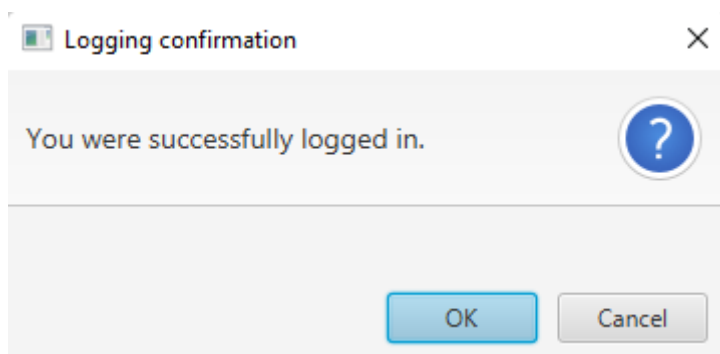
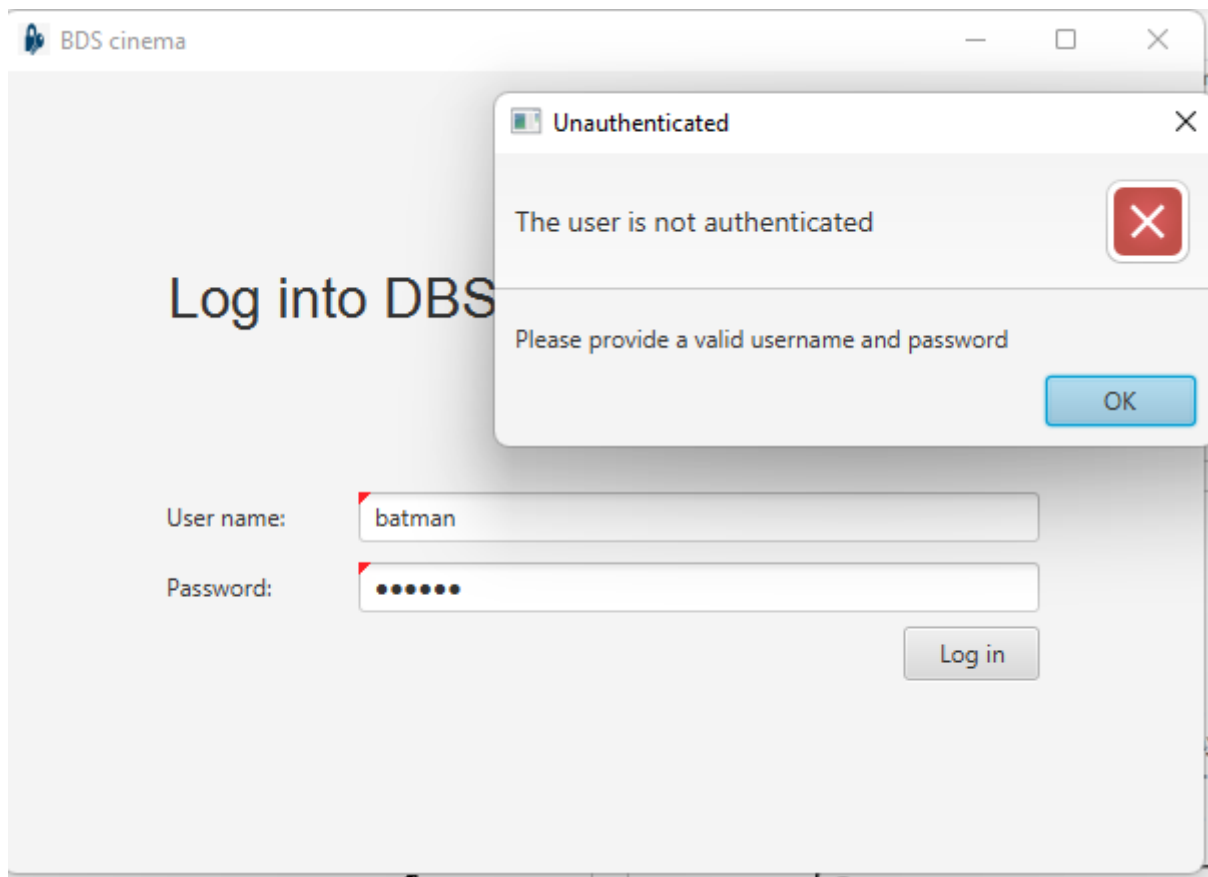
1 usage
public boolean authenticate(String username, String password) {
    if (username == null || username.isEmpty() || password == null || password.isEmpty()) {
        return false;
    }

    PersonAuthView personAuthView = findPersonByUsername(username);
    if (personAuthView == null) {
        throw new ResourceNotFoundException("Provided username is not found.");
    }
    return argon2.verify(personAuthView.getPassword(), password.toCharArray());
}

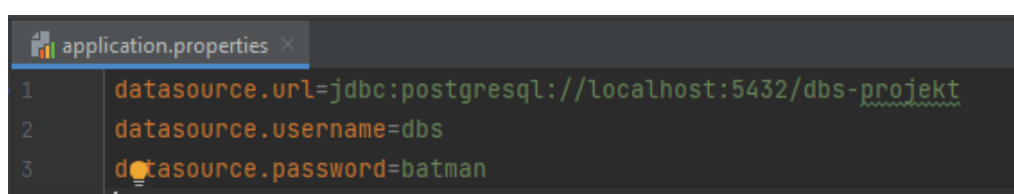
1 usage
final private static Argon2 argon2 = Argon2Factory.create(
    Argon2Factory.Argon2Types.ARGON2id, defaultSaltLength: 16, defaultHashLength: 64
);

```

- The application will have a sign-in window with username/password authentication (if a user enters the wrong credentials, the application will pop up „Username or password is not valid“). This sign-in window won't be any fake login. It must be really validating the username + password from the database (consider loading also user roles – in enterprise systems, the roles during the authentication process are saved to the security context and used later for authorization purposes).



- Create a database user role for the application (not a superuser role!) that can sign in to the database and have suitable privileges. Furthermore, create/use a different than „public“ schema for your database (e.g., the one from the project assignment 2).



GRANT ALL PRIVILEGES ON DATABASE "dbs-projekt" TO dbs;

GRANT USAGE ON SCHEMA bds TO dbs;

GRANT USAGE ON SCHEMA public TO dbs;

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES, TRIGGER, TRUNCATE ON ALL TABLES

IN SCHEMA bds TO dbs;

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES, TRIGGER, TRUNCATE ON ALL TABLES

IN SCHEMA public TO dbs;

- Create CRUD operations (create, read, update, delete) for at least one entity. Do not forget to have it in the GUI (applies to all of the operations). For one entity, implement the findAll operation.

BDS cinema

## Reservations

Enter value to find

Find

Dummy table

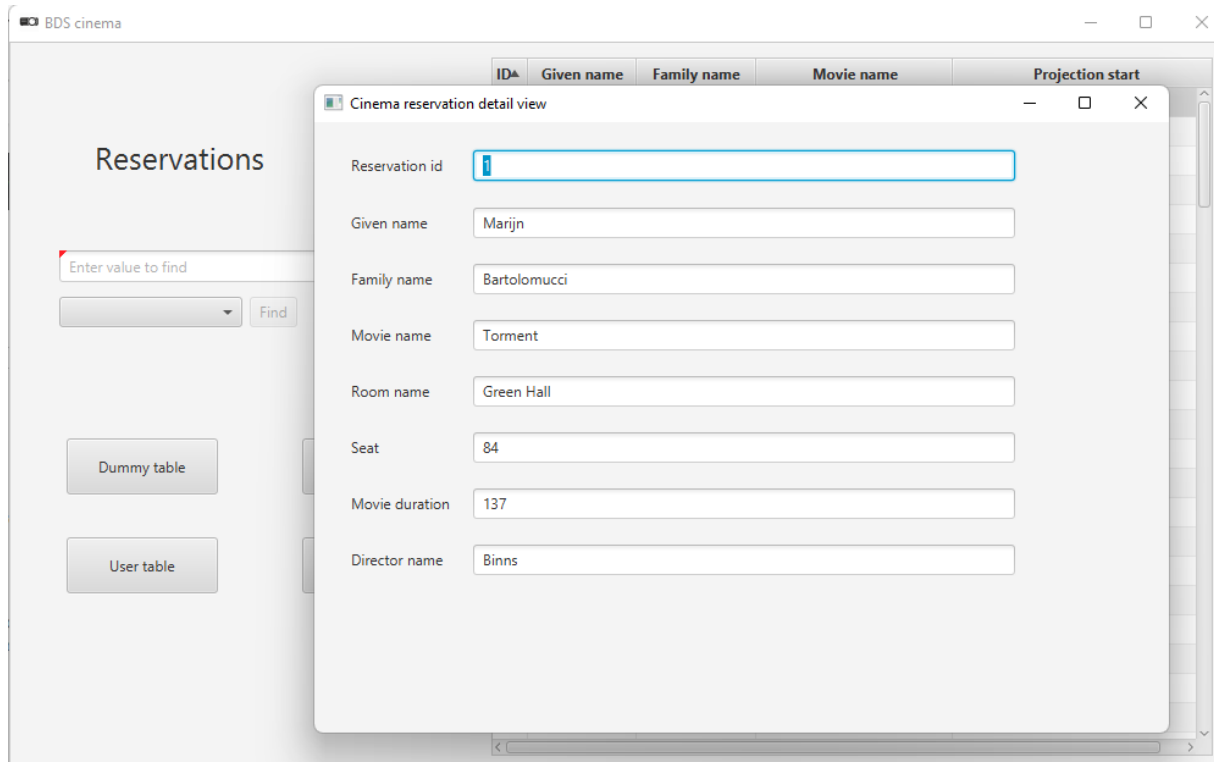
Add reservation

User table

Refresh

ID	Given name	Family name	Movie name	Projection start
1	Marijn	Bartolomucci	Torment	2022-11-26 14:40:00
2	Madelon	Strickett	Holidaz	12-23 10:45:00
3	Claus	Boxell	Men in	10-02 11:40:00
4	Syd	Brock	Holidaz	12-23 10:45:00
5	Monroe	Elacoate	Bedrooms & Hallways	2022-10-12 09:15:00
6	Lelia	Lauxmann	Critters 2: The Main Course	2022-11-16 22:00:00
7	Lelia	Lauxmann	Marva Collins Story, The	2022-10-30 13:50:00
8	Lelia	Lauxmann	Marva Collins Story, The	2022-11-07 09:00:00
9	Bride	Niche	Marva Collins Story, The	2022-11-07 21:00:00
10	Bride	Niche	Marva Collins Story, The	2022-10-30 13:50:00
11	Chrystel	Reichert	Torment	2022-11-26 14:40:00
12	Monroe	Elacoate	Bedrooms & Hallways	2022-10-12 09:15:00
13	Madelon	Strickett	Beauty Shop	2022-09-29 21:30:00
14	Chrystel	Reichert	Marva Collins Story, The	2022-09-14 11:20:00
15	Cissy	Hoyles	Men in White	2022-10-02 11:40:00
16	Zorine	Bust	Beauty Shop	2022-08-30 22:00:00
17	Zorine	Bust	Killer Movie	2022-08-20 12:55:00
18	Zorine	Bust	Holidaze	2022-10-21 11:00:00
19	Syd	Brock	Critters 2: The Main Course	2022-09-24 11:45:00
20	Elisabet	Jochanany	Marva Collins Story, The	2022-09-14 11:20:00
21	Christiane	Swafield	Marva Collins Story, The	2022-10-30 13:50:00
22	Marijn	Bartolomucci	Marva Collins Story, The	2022-11-07 09:00:00

- For one entity, provide a detailed view (you have to use JOIN here).



- Implement one operation in GUI that invokes more than one query and runs them all in one transaction. If something fails, roll back that transaction.

```

public void editProjection(ProjectionEditView projectionEditView) {
    String insertProjectionSQL = "UPDATE bds.reservation SET user_id = ?, projection_id = ?, is_active = ?, reservation_date = ? WHERE reservation_id = ?";
    String checkIfExists = "SELECT r.reservation_id FROM bds.reservation r WHERE r.reservation_id = ? ORDER BY r.reservation_id";
    try (Connection connection = DataSourceConfig.getConnection();
        PreparedStatement preparedStatement = connection.prepareStatement(insertProjectionSQL, Statement.RETURN_GENERATED_KEYS)) {

        preparedStatement.setLong(1, projectionEditView.getUserId());
        preparedStatement.setLong(2, projectionEditView.getProjectionId());
        preparedStatement.setBoolean(3, projectionEditView.getStatus());
        preparedStatement.setTimestamp(4, projectionEditView.getReservation_date());
        preparedStatement.setLong(5, projectionEditView.getId());

        try {
            connection.setAutoCommit(false);
            try (PreparedStatement ps = connection.prepareStatement(checkIfExists, Statement.RETURN_GENERATED_KEYS)) {
                ps.setLong(1, projectionEditView.getId());
                ps.execute();
            } catch (SQLException e) {
                throw new DataAccessException("This reservation to edit do not exists.");
            }

            int affectedRows = preparedStatement.executeUpdate();

            if (affectedRows == 0) {
                throw new DataAccessException("Creating reservation failed, no rows affected.");
            }
            connection.commit();
        } catch (SQLException e) {
            connection.rollback();
        } finally {
            connection.setAutoCommit(true);
        }
    } catch (SQLException e) {
        throw new DataAccessException("Creating reservation operation failed.");
    }
}

```

- [illegible]

- BDS cinema

Reservations

5 OR 1=1

Id

Find

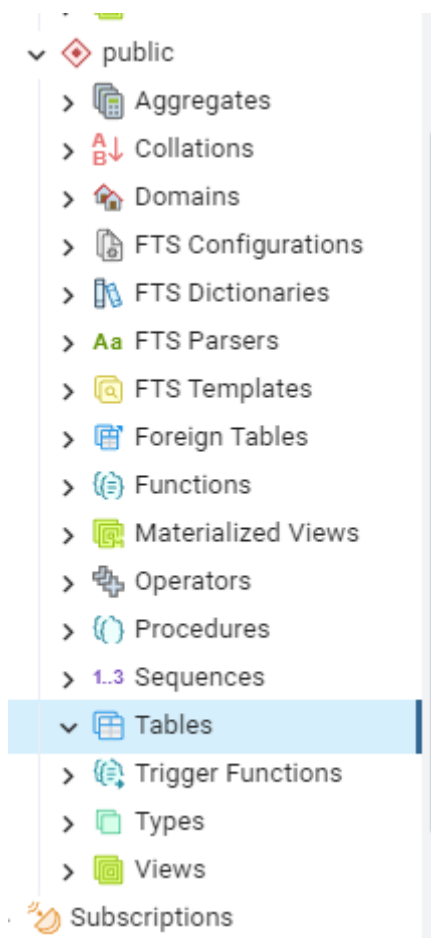
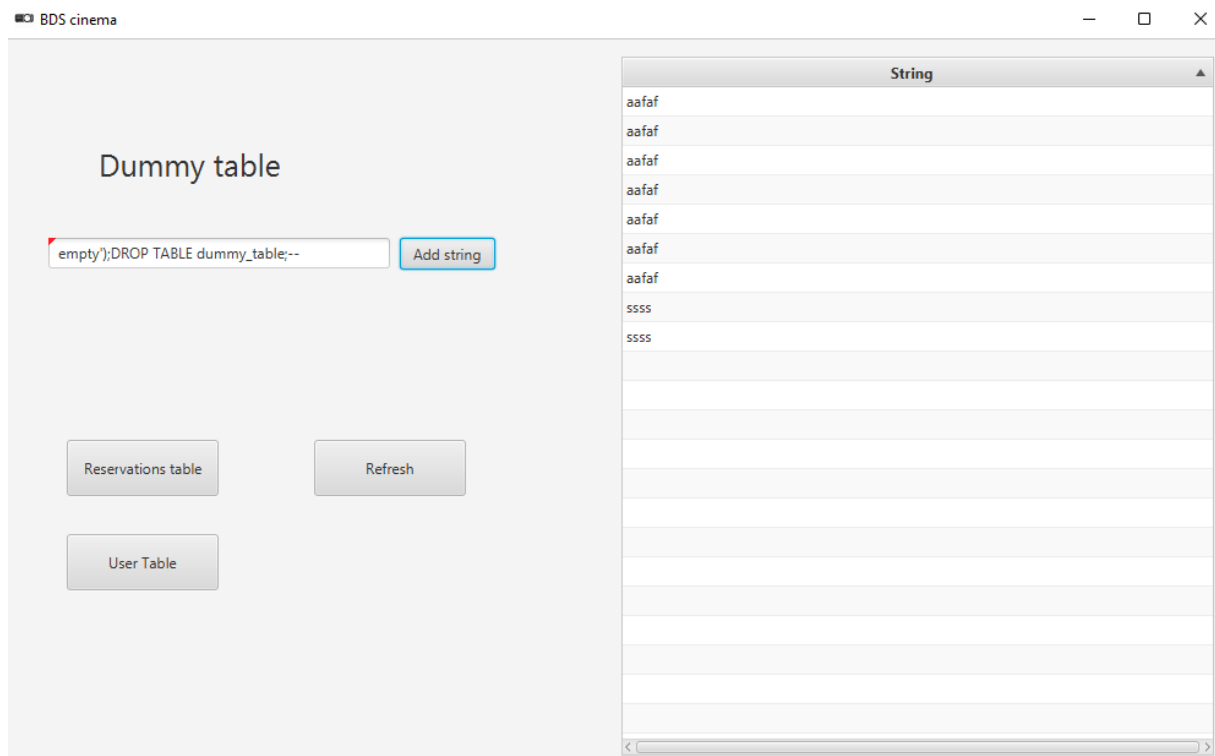
Dummy table

Add reservation

User table

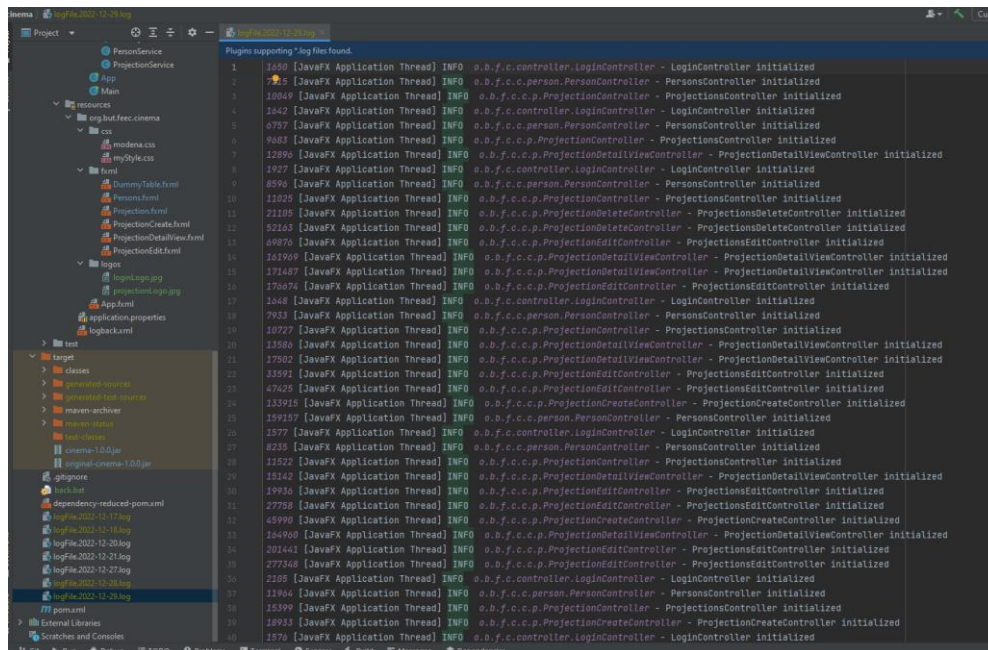
Refresh

ID	Given name	Family name	Movie name	Projection start
1	Marijn	Bartolomucci	Torment	2022-11-26 14:40:00
2	Madelon	Strickett	Holidaze	2022-12-23 10:45:00
3	Claus	Boxell	Men in White	2022-10-02 11:40:00
4	Syd	Brock	Holidaze	2022-12-23 10:45:00
5	Monroe	Elacoate	Bedrooms & Hallways	2022-10-12 09:15:00
6	Lelia	Lauxmann	Critters 2: The Main Course	2022-11-16 22:00:00
7	Lelia	Lauxmann	Marva Collins Story, The	2022-10-30 13:50:00
8	Lelia	Lauxmann	Marva Collins Story, The	2022-11-07 09:00:00
9	Bride	Niche	Marva Collins Story, The	2022-11-07 21:00:00
10	Bride	Niche	Marva Collins Story, The	2022-10-30 13:50:00
11	Chrystel	Reichert	Torment	2022-11-26 14:40:00
12	Monroe	Elacoate	Bedrooms & Hallways	2022-10-12 09:15:00
13	Madelon	Strickett	Beauty Shop	2022-09-29 21:30:00
14	Chrystel	Reichert	Marva Collins Story, The	2022-09-14 11:20:00
15	Cissy	Hoyles	Men in White	2022-10-02 11:40:00
16	Zorine	Bust	Beauty Shop	2022-08-30 22:00:00
17	Zorine	Bust	Killer Movie	2022-08-20 12:55:00
18	Zorine	Bust	Holidaze	2022-10-21 11:00:00
19	Syd	Brock	Critters 2: The Main Course	2022-09-24 11:45:00
20	Elisabet	Jochanany	Marva Collins Story, The	2022-09-14 11:20:00
21	Christiane	Swafield	Marva Collins Story, The	2022-10-30 13:50:00
22	Marijn	Bartolomucci	Marva Collins Story, The	2022-11-07 09:00:00



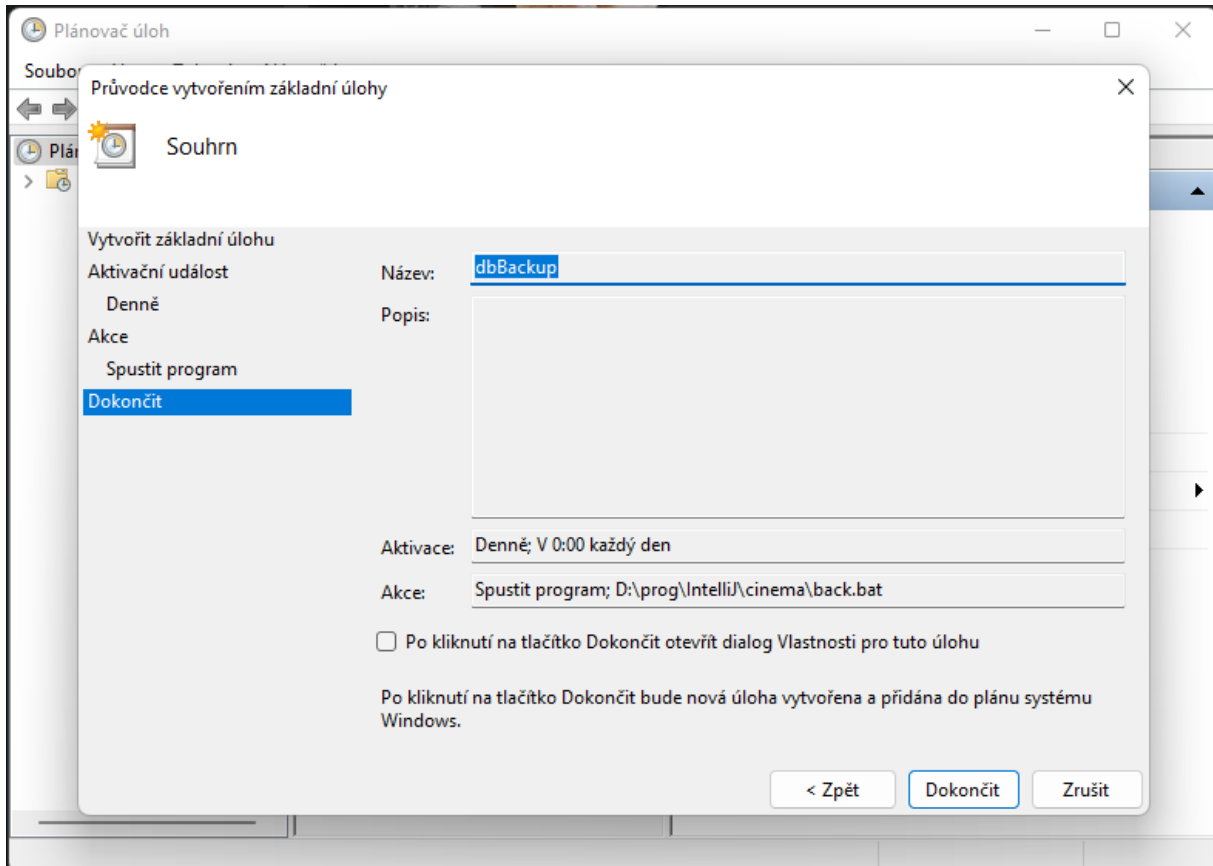
PreparedStatementů zabraňují SQL injection útokům, zamezí zadání kódu do pole.

- Log exceptions suitably (e.g., use SLF4J Logback), set logs archiving per day (avoid log-and-throw antipattern). Avoid Log4j2 older versions < 2.17.1 (as was thrown up in 2021 by CVE 2021-45046, CVE 2021-45105, and CVE-2021-4483).





- Create a script that will back up your database every midnight (**Note:** you can use **pg\_dump**). In practice, more complex way of backups would be used, e.g., <https://www.postgresql.org/docs/current/continuous-archiving.html> to archive the data and logs to the Cloud Object Storages (e.g., Amazon S3)).



```
back.bat
1 @ECHO OFF
2 echo "Starting PostgreSQL backup"
3 set datetime="%DATE:=-%-%TIME::=%"
4 if not exist "backups" md "backups"
5 docker exec -it sedag-db-postgres pg_dump -U dbs dbs-projekt > autobc\postgres-backup-%datetime%.sql
6 echo "Backup created"
7 @ECHO ON
```