

## 6 Anhang

### 6.1 Login (Login.java)

```
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class login extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField email;
```

```
private JPasswordField passwordField;
private JButton anmeldenButton;
private JButton regButton;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                login frame = new login();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */

public login() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(250, 190, 800, 700);
    setResizable(false);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
}
```

```
contentPane.setLayout(null);

JLabel lblNewUserRegister = new JLabel("Login");
lblNewUserRegister.setFont(new Font("Times New Roman", Font.PLAIN, 42));
lblNewUserRegister.setBounds(350, 52, 400, 50);
contentPane.add(lblNewUserRegister);

//email
JLabel lblNewLabel = new JLabel("E-Mail Adresse");
lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblNewLabel.setBounds(242, 243, 160, 50);
contentPane.add(lblNewLabel);

email = new JTextField();
email.setFont(new Font("Tahoma", Font.PLAIN, 32));
email.setBounds(414, 235, 228, 50);
email.setColumns(10);
contentPane.add(email);

//password
JLabel lblPassword = new JLabel("Password");
lblPassword.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblPassword.setBounds(242, 324, 124, 36);
contentPane.add(lblPassword);

passwordField = new JPasswordField();
passwordField.setFont(new Font("Tahoma", Font.PLAIN, 32));
passwordField.setBounds(414, 320, 228, 50);
passwordField.setColumns(10);
contentPane.add(passwordField);
```

```
//anmelden-button
anmeldenButton = new JButton("Anmelden");
anmeldenButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        String password = passwordField.getText();

        String msg = "" + email;
        msg += " \n";

        try {
            Connection connection = DriverManager.getConnection("jdbc:mysql://localhost/foldingpaperstory", "root",
            "");

            //get id from user
            String query = "SELECT ID_User FROM tbl_user WHERE Password = '" + password.hashCode() + "' AND Mail = '" + email.getText() + "' ";
            Statement sta = connection.createStatement();
            ResultSet x = sta.executeQuery(query);

            if (x.next() == false) {
                JOptionPane.showMessageDialog(anmeldenButton, "Passwort oder E-Mail Adresse Falsch");
            } else {
                String id = x.getString(x.findColumn("ID_User"));
```

```
//check if last expand was 2 hours ago
query = "SELECT timestamp FROM tbl_userstory WHERE FK_User = '" + id + "' ORDER BY ID_StoryLine
DESC";

sta = connection.createStatement();
ResultSet l = sta.executeQuery(query);
if (l.next() == true){

    LocalDateTime startTime =
LocalDateTime.parse(l.getString(l.findColumn("timestamp")),DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"));

    LocalDateTime now = LocalDateTime.now();
    long difference = ChronoUnit.HOURS.between(startTime, now);

    if (difference>= 2){

        expandhistory frame = new expandhistory(id);
        frame.setVisible(true);
        dispose();

    }else{

        long wait = 2 - difference;
        time frame = new time(wait);
        frame.setVisible(true);
        dispose();
    }
}else{

    expandhistory frame = new expandhistory(id);
```

```
        frame.setVisible(true);
        dispose();}

    }
    connection.close();
} catch (Exception exception) {
    exception.printStackTrace();
}
}

});
anmeldenButton.setFont(new Font("Tahoma", Font.PLAIN, 22));
anmeldenButton.setBounds(320, 600, 150, 30);
contentPane.add(anmeldenButton);

//reg-Button
regButton = new JButton("Registrieren");
regButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            register frame = new register();
            frame.setVisible(true);
            dispose();

        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
});
regButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
        regButton.setBounds(600, 10, 150, 30);  
        contentPane.add(regButton);  
  
    }  
}
```

## 6.2 Registration (register.java)

```
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class register extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField email;
    private JPasswordField passwordField;
    private JButton registrierButton;
    private JButton anButton;

    /**
     * Launch the application.
     */
}
```



```
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                register frame = new register();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */

public register() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(250, 190, 800, 700);
    setResizable(false);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewUserRegister = new JLabel("Registrierung");
    lblNewUserRegister.setFont(new Font("Times New Roman", Font.PLAIN, 42));
    lblNewUserRegister.setBounds(300, 52, 400, 50);
    contentPane.add(lblNewUserRegister);
}
```

```
//email
JLabel lblNewLabel = new JLabel("E-Mail Adresse");
lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblNewLabel.setBounds(242, 243, 160, 50);
contentPane.add(lblNewLabel);

email = new JTextField();
email.setFont(new Font("Tahoma", Font.PLAIN, 32));
email.setBounds(414, 235, 228, 50);
email.setColumns(10);
contentPane.add(email);

//password
JLabel lblPassword = new JLabel("Password");
lblPassword.setFont(new Font("Tahoma", Font.PLAIN, 20));
lblPassword.setBounds(242, 324, 124, 36);
contentPane.add(lblPassword);

passwordField = new JPasswordField();
passwordField.setFont(new Font("Tahoma", Font.PLAIN, 32));
passwordField.setBounds(414, 320, 228, 50);
passwordField.setColumns(10);
contentPane.add(passwordField);

//reg-button
registrierButton = new JButton("Registrieren");
registrierButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```

```

String emailId = email.getText();
String password = passwordField.getText();
int hashpassword = password.hashCode();

try {

    Connection connection = DriverManager.getConnection("jdbc:mysql://localhost/foldingpaperstory", "root",
    "");

    //Check if E-Mail input is valid

    //pattern
    final String EMAIL_PATTERN =
    "^[_A-Za-z0-9-\\+](\\.[_A-Za-z0-9-]+)*@"
    + "[A-Za-z0-9-](\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$";

    final String passwort_pattern =
    "([A-Za-z0-9\\?!\\.,-]+)";

    //check if Mail is already used
    String query = "SELECT Mail FROM tbl_user WHERE Mail = '" + email.getText() + "' ";

    Statement sta = connection.createStatement();
    ResultSet res = sta.executeQuery(query);

    if (!emailId.matches(EMAIL_PATTERN) || res.next() == true || !password.matches(passwort_pattern) ||
password.length() < 8 ) {
        JOptionPane.showMessageDialog(registrierButton, "E-Mail Adresse wird entweder bereits gebraucht
oder Eingaben entsprechen nicht den vorgaben");
    }
}

```

```
    }else{

        query = "INSERT INTO tbl_user (Password,Mail,Rolle) values('" + hashpassword + "', '" + emailId + "',
0)";

        sta = connection.createStatement();
        int x = sta.executeUpdate(query);

        if (x == 0) {
            JOptionPane.showMessageDialog(registrierButton, "This is alredy exist");
        } else {

            //get id from user
            query = "SELECT ID_User FROM tbl_user WHERE Password = '" + hashpassword + "' AND Mail = '" +
emailId + "' ";

            sta = connection.createStatement();
            ResultSet ids = sta.executeQuery(query);
            ids.next();
            String id = ids.getString(ids.findColumn("ID_User"));

            expandhistory frame = new expandhistory(id);
            frame.setVisible(true);
            dispose();
        }
    }
}
```

```
        connection.close();
    }
} catch (Exception exception) {
    exception.printStackTrace();
}
}

});
registrierButton.setFont(new Font("Tahoma", Font.PLAIN, 22));
registrierButton.setBounds(320, 600, 150, 30);
contentPane.add(registrierButton);

//an-Button
anButton = new JButton("Anmelden");
anButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            login frame = new login();
            frame.setVisible(true);
            dispose();

        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
});
anButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
anButton.setBounds(600, 10, 150, 30);
```

```
        contentPane.add(anButton);  
    }  
}
```

## 6.3 Geschichte erweitern (expandhistory.java)

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import javax.swing.Timer;

public class expandhistory extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField s1;
    private JTextField s2;
    private JButton sendenButton;
    private JButton abmeldenButton;

    /**
```

```
* Launch the application.
*/
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                expandhistory frame = new expandhistory("1");
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */

public expandhistory(String ID) {

    final String SentenceInput_Pattern =
        "([A-Za-z0-9\\?! ,.-]+)";

    String id = ID;
    try {
        Connection connection = DriverManager.getConnection("jdbc:mysql://localhost/foldingpaperstory", "root", "");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```



```
setBounds(250, 190, 800, 700);
setResizable(false);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(null);

//abfrage vorheriger Satz
String query = "SELECT sentence2 FROM `tbl_story` ORDER BY ID_StoryLine DESC";

Statement sta = connection.createStatement();
ResultSet x = sta.executeQuery(query);

if (x.next() == false) {
    JLabel lblsmattittle = new JLabel("Schreibe die ersätzen Sätze der Geschichte...");
    lblsmattittle.setFont(new Font("Times New Roman", Font.PLAIN, 30));
    lblsmattittle.setBounds(350, 52, 400, 50);
    contentPane.add(lblsmattittle);
} else {

    String s2 = x.getString(x.findColumn("sentence2"));
    JLabel lblsmattittle = new JLabel(s2);
    lblsmattittle.setFont(new Font("Times New Roman", Font.PLAIN, 30));
    lblsmattittle.setBounds(350, 52, 400, 50);
    contentPane.add(lblsmattittle);

}

JLabel lbltitle = new JLabel("Schreibe weiter...");
lbltitle.setFont(new Font("Times New Roman", Font.PLAIN, 42));
lbltitle.setBounds(300, 152, 400, 50);
```

```
contentPane.add(lbltitle);

//s1
JLabel s1label = new JLabel("Satz 1");
s1label.setFont(new Font("Tahoma", Font.PLAIN, 20));
s1label.setBounds(242, 243, 160, 50);
contentPane.add(s1label);

s1 = new JTextField(2);
s1.setFont(new Font("Tahoma", Font.PLAIN, 32));
s1.setBounds(414, 235, 228, 50);
s1.setColumns(10);
contentPane.add(s1);

//s2
JLabel s2label = new JLabel("Satz 2");
s2label.setFont(new Font("Tahoma", Font.PLAIN, 20));
s2label.setBounds(242, 324, 124, 36);
contentPane.add(s2label);

s2 = new JTextField();
s2.setFont(new Font("Tahoma", Font.PLAIN, 32));
s2.setBounds(414, 320, 228, 50);
s2.setColumns(10);
contentPane.add(s2);

//counter
```

```
JLabel counter = new JLabel("Du hast nur 2 min!!");
counter.setFont(new Font("Tahoma", Font.PLAIN, 20));
counter.setForeground(Color.RED);
counter.setBounds(300, 400, 400, 50);
contentPane.add(counter);
Timer timer = new Timer(2*60*1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        login frame = new login();
        frame.setVisible(true);
        dispose();
    }
});

timer.setRepeats(false); // Only execute once
timer.start();

System.out.println(timer.getDelay()/1000/60 + "min");

//anmelden-button
sendenButton = new JButton("senden");
} catch (Exception exception) {
    exception.printStackTrace();
}
```

```

sendenButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if (!s2.getText().matches(SentenceInput_Pattern) || !s1.getText().matches(SentenceInput_Pattern) ||
s2.getText().length() > 25 || s1.getText().length() > 25 ){
            JOptionPane.showMessageDialog(sendenButton, "Eingabe invalide");
        }else{

            try {
                Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/foldingpaperstory?allowMultiQueries=true", "root", "");

                String sent1 = s1.getText();
                String sent2 = s2.getText();

                //abspeichern der Sätze
                String query = "INSERT INTO tbl_story (sentence1,sentence2) values('" + sent1 + "','" + sent2
+ "');INSERT INTO `tbl_userstory` (`ID_StoryLine`, `timestamp`, `FK_User`, `FK_StoryLine`) VALUES (NULL, NOW(),'" + id +
"', (SELECT ID_StoryLine FROM `tbl_story` WHERE sentence1 = '" + sent1 + "' AND sentence2 = '" + sent2 + "') )";

                Statement sta = connection.createStatement();
                int x = sta.executeUpdate(query);
                if (x == 0) {
                    JOptionPane.showMessageDialog(sendenButton, "konnte Geschichte nicht erweitern, versuche es
erneut.");
                } else {
                    downloadhistory frame = new downloadhistory(id);
                    frame.setVisible(true);

```

```
        dispose();
    }
    connection.close();
} catch (Exception exception) {
    exception.printStackTrace();
}
}
});

sendenButton.setFont(new Font("Tahoma", Font.PLAIN, 22));
sendenButton.setBounds(320, 600, 150, 30);
contentPane.add(sendenButton);

//reg-Button
abmeldenButton = new JButton("abmelden");
abmeldenButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            login frame = new login();
            frame.setVisible(true);
            dispose();

        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
});
abmeldenButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
abmeldenButton.setBounds(600, 10, 150, 30);
contentPane.add(abmeldenButton);

    }
}
```

#### 6.4 Geschichte herunterladen (downloadhistory.java)

```
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.charset.Charset;

public class downloadhistory extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField s1;
    private JTextField s2;
    private JButton downloadButton;
    private JButton weiterButton;
    private JButton abmeldenButton;
```

```
/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                downloadhistory frame = new downloadhistory("0");
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */

public downloadhistory(String ID) {
    String id = ID;
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(250, 190, 800, 700);
    setResizable(false);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lbltitle = new JLabel("Danke \n für deine \n\n Ergänzung :");
```



```
lbltitle.setFont(new Font("Times New Roman", Font.PLAIN, 32));
lbltitle.setBounds(200, 52, 400, 400);
contentPane.add(lbltitle);

//download-button
downloadButton = new JButton("Geschichte downloaden");
downloadButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        //Todo: herunterladen geschichte in txt

        try {
            Connection connection = DriverManager.getConnection("jdbc:mysql://localhost/foldingpaperstory", "root",
""");

            //check what rolle
            String query = "SELECT Rolle FROM tbl_user WHERE ID_User = '" + id + "' ";
            Statement sta = connection.createStatement();
            ResultSet r = sta.executeQuery(query);
            r.next();
            String rolle = r.getString("Rolle");
            System.out.println(rolle);

            if (rolle.matches("0")){
                query = "SELECT sentence1, sentence2 FROM tbl_story";
            }
            else{ query = "SELECT `tbl_story`.`sentence1`, `tbl_story`.`sentence2`, `tbl_user`.`Mail` FROM
`tbl_story` LEFT JOIN `tbl_userstory` ON `tbl_userstory`.`FK_StoryLine` = `tbl_story`.`ID_StoryLine` LEFT JOIN `tbl_user`
ON `tbl_userstory`.`FK_User` = `tbl_user`.`ID_User`"; }
```

```
sta = connection.createStatement();
ResultSet x = sta.executeQuery(query);

if (x.next() == false) {
    JOptionPane.showMessageDialog(downloadButton, "Die Geschichte hat noch keine Inhalt. Es wurden
keine Sätze gefunden");
} else {

    //create output file
    try {
        String home = System.getProperty("user.home");
        FileWriter myWriter = new FileWriter(home + "/Downloads/Geschichte.txt",
Charset.forName("UTF8"));

        if (rolle.matches("0")){
            myWriter.write(x.getString("sentence1") + " ");
            myWriter.write(x.getString("sentence2") + "\n");

            while (x.next()){

                myWriter.write(x.getString("sentence1") + " ");
                myWriter.write(x.getString("sentence2") + "\n");

            }
        }else{
            myWriter.write(x.getString("sentence1") + " ");
            myWriter.write(x.getString("sentence1") + " ");
            myWriter.write(x.getString("Mail") + "\n");

            while (x.next()){
```

```
        myWriter.write(x.getString("sentence1") + " ");
        myWriter.write(x.getString("sentence1") + " ");
        myWriter.write(x.getString("Mail") + "\n");

    }

}

myWriter.close();
JOptionPane.showMessageDialog(downloadButton, "Die Geschichte wurde heruntergeladen");

} catch (IOException exception) {
    System.out.println("An error occurred.");
    exception.printStackTrace();
}

}
connection.close();
} catch (Exception exception) {
    exception.printStackTrace();
}

}

});
downloadButton.setFont(new Font("Tahoma", Font.PLAIN, 22));
downloadButton.setBounds(260, 450, 280, 30);
contentPane.add(downloadButton);

//weiter-Button
weiterButton = new JButton("weiter");
weiterButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```

```
        time frame = new time(2);
        frame.setVisible(true);
        dispose();
    }
});
weiterButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
weiterButton.setBounds(320, 600, 150, 30);
contentPane.add(weiterButton);

//reg-Button
abmeldenButton = new JButton("abmelden");
abmeldenButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            register frame = new register();
            frame.setVisible(true);

        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
});
abmeldenButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
abmeldenButton.setBounds(600, 10, 150, 30);
contentPane.add(abmeldenButton);

}
```

```
}
```

## 6.5 Zeit Counter (time.java)

```
import java.awt.EventQueue;
```

```
import java.awt.Font;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.border.EmptyBorder;
```

```
public class time extends JFrame {  
    private static final long serialVersionUID = 1L;  
    private JPanel contentPane;  
    private JButton weiterButton;  
    private JButton abmeldenButton;
```

```
    /**
```

```
     * Launch the application.
```

```
     */
```

```
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    time frame = new time(0);  
                    frame.setVisible(true);  
                } catch (Exception e) {
```

```
        e.printStackTrace();
    }
}

/**
 * Create the frame.
 */

public time(long Diff) {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(250, 190, 800, 700);
    setResizable(false);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    //abfrage der Zeit
    JLabel lbltitle = new JLabel("Schreibe in " + Diff + " h weiter");
    lbltitle.setFont(new Font("Times New Roman", Font.PLAIN, 32));
    lbltitle.setBounds(200, 52, 400, 400);
    contentPane.add(lbltitle);

    //weiter-Button
    weiterButton = new JButton("fertig");
    weiterButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
```

```
        dispose();
    }
});
weiterButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
weiterButton.setBounds(320, 600, 150, 30);
contentPane.add(weiterButton);

//reg-Button
abmeldenButton = new JButton("abmelden");
abmeldenButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            register frame = new register();
            frame.setVisible(true);
            dispose();

        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
});
abmeldenButton.setFont(new Font("Tahoma", Font.PLAIN, 20));
abmeldenButton.setBounds(600, 10, 150, 30);
contentPane.add(abmeldenButton);

}
}
```

## 6.6 Struktur Datenbank (db\ddl.sql)

/\*

Author: Lara Felix

Titel: M223 - FoldingPaperStory

\*/

-- DATABASE

DROP DATABASE IF EXISTS FoldingPaperStory;

CREATE DATABASE FoldingPaperStory;

USE FoldingPaperStory;

-- CREATE TABLES

CREATE TABLE tbl\_User (

    ID\_User INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

    Mail VARCHAR(50),

    Password VARCHAR(50),

    Rolle Int

);

CREATE TABLE tbl\_Story (



```
ID_StoryLine INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
sentence1 VARCHAR(100),  
sentence2 VARCHAR(100)  
);
```

```
CREATE TABLE tbl_UserStory (  
ID_StoryLine INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
timestamp DATE NOT NULL,  
FK_User INT UNSIGNED,  
FK_StoryLine INT UNSIGNED,  
FOREIGN KEY (FK_User) REFERENCES tbl_User(ID_User),  
FOREIGN KEY (FK_StoryLine) REFERENCES tbl_Story(ID_StoryLine)  
);
```

## 6.7 Daten Datenbank (db\import.sql)

```
/*
```

```
Author: Lara Felix
```

```
Titel: M223 - FoldingPaperStory
```

```
*/
```

```
USE FoldingPaperStory;
```

```
DROP TABLE IF EXISTS tbl_user;
```

```
/*
```

```
writer
```

```
*/
```

```
INSERT INTO `tbl_user` (`ID_User`, `Mail`, `Password`, `Rolle`)
```

```
VALUES (
```

```
NULL, 'testing@user.ch', '745064826', '0'
```

```
);
```

/\*

gamemaster

\*/

INSERT INTO `tbl\_user` (`ID\_User`, `Mail`, `Password`, `Rolle`)

VALUES (

NULL, 'game@master.ch', '-1888845021', '1'

);