



# **Dokumentation Software-Projekt im Modul Aktuelles Thema der Eingebetteten Systeme**

Design, Implementation und Testung einer  
Arbeitszeiterfassungssoftware

**Samuel Hessberger**

samuel-lukas.hessberger@et.hs-fulda.de

**Louis Försch**

louis-henry.foersch@et.hs-fulda.de

## Inhalt

Pflichtenheft .....	2
Benutzertest: .....	4
Software-Architektur .....	5
Software-Design .....	6
Testfälle Arbeitszeitmanagement .....	11

# Pflichtenheft

## 1. Gegenstand und Kontext

### a. Auftraggeber

Florian Schäfer im Rahmen des Moduls „Aktuelles Thema der Eingebetteten Systeme 1 – ET5062SU“

### b. Auftragnehmer

Gruppe 2, Louis Försch (1445659), Samuel Hessberger (1219689)

### c. Gegenstand

Es soll eine Arbeitszeiterfassungssoftware mit Benutzeroberfläche erstellt werden. Diese soll Initialisierungsdateien Einlesen und Auswerten können. Außerdem sollen verschiedene Mitarbeiterinnen und Mitarbeiter mit automatischer Berechnung der Arbeitszeit gespeichert werden können.

## 2. Anforderungen

### a. Funktionale Anforderungen

- i. Die Benutzeroberfläche soll die aktuelle Uhrzeit anzeigen
- ii. Der Benutzer soll sich an- und abmelden können
- iii. Der Benutzer soll für die Anmeldung seine ID eingeben können
- iv. Die Software soll Mitarbeiterinformation ausgeben
- v. Die Software soll Init-Dateien lesen und verarbeiten können
- vi. CSV-Dateien lesen und speichern können
- vii. Die Software soll definierte Pausenzeiten einstellen können
- viii. Die Arbeitszeit soll nachträglich gerundet werden
- ix. Der Mitarbeiter soll sein Urlaub oder Überstundenabbau beantragen können
- x. Der Mitarbeiter soll den Status über seine Anträge abfragen können
- xi. Aus der Arbeitssoftware sollen die Einträge der Initialisierungsdatei angepasst werden können
- xii. Die Software soll eine Abmeldung ermöglichen
- xiii. Beim Löschen eines Mitarbeiters sollen seine Daten archiviert werden
- xiv. Die Software soll das An- und Abmelden eines Arbeitnehmers unbegrenzt oft zulassen
- xv. Die Software soll einen autorisierten Zugang bereitstellen
  1. Dem autorisierten Zugang sollen sämtliche Anträge (Urlaubsantrag und Überstundenabbau) angezeigt werden
  2. Der autorisierte Nutzer soll Krankheitstage eintragen können
  3. In der Benutzerverwaltung soll ein neuer Benutzer angelegt, gelöscht oder verändert werden können
- xvi. Das Passwort des autorisierten Zugangs soll in der Software anpassbar sein
- xvii. Es soll eine gewünschte Bilddatei angezeigt werden können
- xviii. Die Software soll einen Button für die Benutzerverwaltung haben

### b. Nicht-Funktionale Anforderungen

- i. Die Benutzeroberfläche soll die Uhrzeit minutengenau anzeigen
- ii. Die Software soll Feedback geben, wenn die Eingabe erfolgreich war
- iii. Die ID soll aus einer 6-stelligen Zahlenfolge bestehen

- iv. Die Mitarbeiter-Dateien sollen in einer übersichtlichen Ordnerstruktur abgelegt werden (Abteilung -> Name -> Jahr -> CSV-Datei, Init-Datei?)
- v.
  - 1. Mitarbeiterinformationen sollen Name, Resturlaub und Überstunden beinhalten
  - 2. (Die Mitarbeiterinformationen sollen für einen einstellbaren Zeitraum angezeigt werden)
- vi.
  - 1. Die CSV-Datei soll eine Zeile mit Überschriften besitzen
  - 2. (Die Pfade zu den CSV-Dateien sollen im Nachhinein anpassbar sein)
- vii.
  - 1. Die Pausenzeiten sollen nur durch den autorisierten Zugang dynamisch oder zu festen Uhrzeiten vorgegeben werden können
- viii. Die Rundung der Arbeitszeit soll in 1, 5 oder 15 Minuten-Schritten erfolgen. Dies soll vom autorisierten Profil angepasst werden können
- ix. Der Antrag soll durch autorisiertes Benutzerprofil genehmigt werden können

### **3. Randbedingungen**

#### **a. Technische Randbedingungen**

- i. Entwicklungsumgebung: Visual Studio Community
- ii. Frameworks: .NET inklusive WPF
- iii. Programmiersprache C#

#### **b. Randbedingungen der Erstellung**

Dieses Software-Projekt wird im Rahmen des Moduls „Aktuelles Thema der Eingebetteten Systeme 1 – ET5062SU“ bei Florian Schäfer bearbeitet. Dieses Projekt findet im Sommersemester 2023 statt. Das Modul besteht aus vier Semesterwochenstunden, bei welchen zwei Stunden der Vorlesung dienen und die anderen zwei Stunden der Projektbearbeitung.

### **4. Fortschritt + Planung**

- a. **M1:** Fertigstellung Pflichtenheft am 05.05.2023
- b. **M2:** Architektur und Testfälle am 12.05.2023
- c. **M3:** Design der Oberfläche am 19.05.2023
- d. **M4:** Implementierung am 09.06.2023
- e. **M5:** Systemtest am 23.06.2023
- f. **M6:** Dokumentation und Präsentation am 07.07.2023

**Autoren: Louis Försch & Samuel Hessberger**

## Benutzertest:

**Führen Sie die folgenden Aufgaben nacheinander aus:**

Schritt 1: Gehen Sie in das Menü „Zeiterfassung“. i.O.

Schritt 2: Melden Sie sich mit der ID: 123456 an. i.O.

Schritt 3: Ändern Sie ihren Status über den Button „kommen / Gehen“. i.O. (Weniger Infos bei der Fragestellung)

Schritt 4: Melden Sie sich ab. i.O.

Schritt 5: Melden Sie sich mit der ID: 123456 und dem Passwort: 123456 unter dem Button „Anmelden“ an. i.O. (Weniger Infos bei der Fragestellung)

Schritt 6: Lassen Sie sich die Mitarbeiterinformationen ausgeben und schauen Sie nach wie viel Urlaub Sie noch zur Verfügung haben. i.O.

Schritt 7: Gehen Sie auf den Reiter „Mitarbeiteranträge“ und erstellen Sie einen neuen Antrag.

- Typ: Urlaub
- Startdatum: 30.06.2023
- Enddatum: 14.07.2023 i.O.

Schritt 8: Melden Sie sich ab. i.O.

Schritt 9: Melden Sie sich nun mit folgender ID: 123123 und folgendem Passwort: 654321 an. i.O.

Schritt 10: Navigieren Sie zu dem Menü „Mitarbeiteranträge“ und nehmen Sie den eben erstellten Antrag von Rainer Zufall an. i.O. (Weniger Infos bei der Fragestellung)

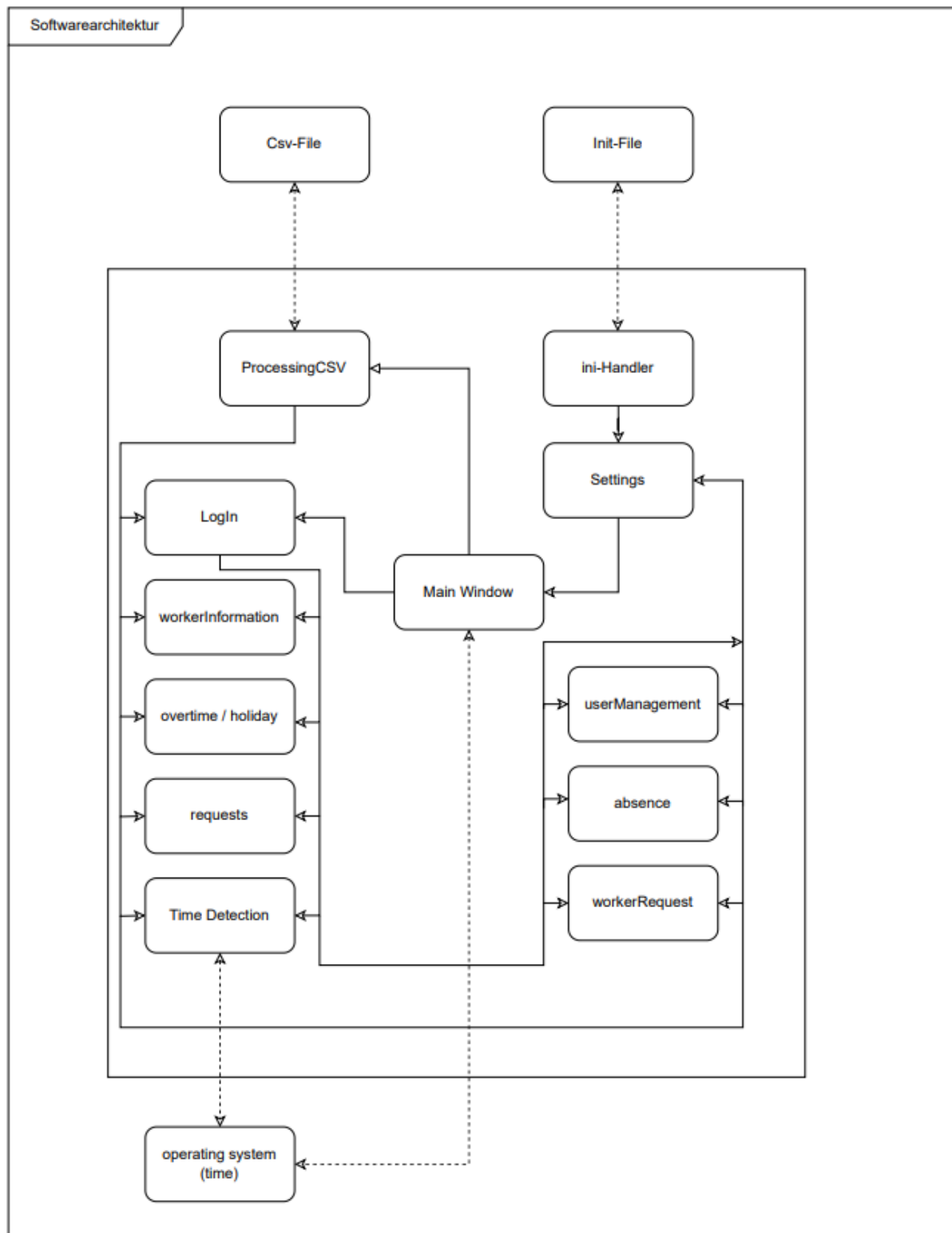
Schritt 11: Navigieren Sie zum Menü „Benutzerverwaltung“ und legen Sie einen Benutzer an mit beliebigen Vor- und Nachnamen, welcher in Fulda wohnt und 48 Jahre alt ist. i.O. (Weniger Infos bei der Fragestellung)

Schritt 12: Bestätigen Sie die Eingaben. i.O.

Schritt 13: Bearbeiten Sie den eben erstellten Benutzer, indem Sie ihm einen neuen Vor- und/oder Nachnamen geben. i.O.

Schritt 14: Löschen Sie den eben erstellten Mitarbeiter. i.O.

## Software-Architektur



## Software-Design

### ProcessingCSV

```
/**
```

```
 * Ergebnis der Überprüfung
```

```
 */
```

```
public enum LoginResult;
```

```
{
```

```
    IDNotFound,
```

```
    PWDIncorrect,
```

```
    UserCorrect,
```

```
    AdminCorrect,
```

```
    TimeDetectionIdFound,
```

```
    TimeDetectionIdNotFound
```

```
}
```

```
/**
```

```
 * Gleicht Benutzername und Passwort mit der Datenbank ab
```

```
 * @param userID: Die eingegebene ID
```

```
 * @param userPWD: Das eingegebene Passwort
```

```
 * @retval: einen int, welcher auf den Enum LoginResult verweist
```

```
 */
```

```
public static int checkLogin (string userID, string userPWD);
```

```
/**
```

```
 * Der neue Benutzer soll mit Namen, ID, Passwort, Geburtsdatum und Wohnort einem Verzeichnis
```

```
 * hinzugefügt werden
```

```
 * @param ID: Die neue ID
```

```
 * @param pwd: Das Passwort
```

```
 * @param dateOfBirth: Das Geburtsdatum des Nutzers
```

```
 * @param residence: Der Wohnort des Nutzers
```

```
 */
```

```
public static void addUserToFile(string ID, string pwd, string dateOfBirth, string residence);
```

```
/**
```

```
 * Ein Benutzer soll mit soll bearbeitet werden können
```

```

* @param pwd: Das neue Passwort
* @param dateOfBirth: Das veränderte Geburtsdatum des Nutzers
* @param residence: Der veränderte Wohnort des Nutzers
*/

public static void editUser(string pwd, string dateOfBirth, string residence);

/**
* Holt die Mitarbeiterinformationen eines bestimmten Mitarbeiters aus der CSV-Datei
* @param id: Die ID des Mitarbeiters
*/

public static string getWorkerInformation (string id);

/**
* Verschiebt den Ordner des Mitarbeiters in ein Archiv, sodass dieser sich nicht mehr anmelden kann
* @param id: Die ID des zu löschenden Mitarbeiters
*/

public static void deleteUser (string id);

/**
* Trägt die Uhrzeit in die CSV ein, zu welcher sich der Arbeitnehmer in der Zeiterfassung als
* „Anwesend“ eingetragen hat
* @param id: Die ID des Mitarbeiters
* @param CheckIn: Uhrzeit, bei welcher der Mitarbeiter eingecheckt ist
*/

public static void userCheckedIn (string id, DateTime CheckIn);

/**
* Trägt die Uhrzeit in die CSV ein, zu welcher sich der Arbeitnehmer in der Zeiterfassung als
* „Abwesend“ eingetragen hat
* @param id: Die ID des Mitarbeiters
* @param CheckOut: Uhrzeit, bei welcher der Mitarbeiter sich aus der Zeiterfassung als „abwesend“
* gemeldet hat
*/

public static void userCheckedOut (string id, DateTime CheckIn);

```



## Login

/\*\*

\* leitet die Eingaben an die ID und Passwort Überprüfung weiter und gibt Fehlermeldungen bei

\* falscher Eingabe aus

\*/

**private void btnLogin(object sender, RoutedEventArgs e);**

/\*\*

\* leitet die Eingaben an die ID und Passwort Überprüfung weiter und gibt Fehlermeldungen bei

\* falscher Eingabe aus

\*/

**private void btnLogin(object sender, RoutedEventArgs e);**

## TimeDetection

/\*\*

\* gibt die aktuelle Uhrzeit bei An- und Abmeldung an die ProcessingCSV weiter. Außerdem wird in

\* einer CSV-Datei vermerkt, dass dieser Mitarbeiter an- oder abgemeldet ist

\*/

**private void btnCheckInOut(object sender, RoutedEventArgs e);**

## Init-Handler

/\*\*

\* Diese Funktion liest die init-Datei aus

\*/

**private static string Read (object sender, RoutedEventArgs e);**

/\*\*

\* Diese Funktion schreibt in die init-Datei

\*/

**private static string Write (object sender, RoutedEventArgs e);**

## Main-Window

/\*\*

\* Liest die Init-Werte aus der Init-Datei ein

\*/

**private void getInitVals ();**

/\*\*

\* Liest die Systemzeit aus

\*/

**private void getTime(object sender, EventArgs e)**

## **userManagement**

/\*\*

\* Daten des neuen Mitarbeiters werden an ProcessingCSV übergeben, indem die Funktionen

\* aufgerufen werden

\*/

**private void newUser()**

/\*\*

\* Die ID des zu löschenden Users wird an ProcessingCSV weitergeleitet

\* @param id: Die ID des Mitarbeiters, welcher gelöscht werden soll

\*/

**private void deleteUser(string id)**

/\*\*

\* Die ID des zu bearbeitenden Users wird an ProcessingCSV weitergeleitet

\* @param id: Die ID des Mitarbeiters, welcher bearbeitet werden soll

\*/

**private void editUser(string id)**

/\*\*

\* Die Liste mit den angelegten Mitarbeitern füllen

\*/

**private void fillListBox()**

## **absence**

/\*\*

\* Die Liste wird mit den aktuellen Abwesenheiten der Mitarbeiter gefüllt

\*/

**private void fillListBox()**

## **Settings**

/\*\*

\* Die veränderten Einstellungen werden ausgelesen und an den Init-Handler übergeben

\*/

**private void getModifiedSettings();**

## **Requests**

/\*\*

\* Legt eine neue Anfrage an

\*/

**private void newRequest();**

/\*\*

\* Löscht eine nicht-genehmigte Anfrage

\*/

**private void deleteRequest();**

## **Requests**

/\*\*

\* Nimmt den Antrag eines Arbeitnehmers an oder lehnt diesen ab

\*/

**private void acceptOrRecline();**

## Testfälle Arbeitszeitmanagement

Beschreibung	Testfall	Soll-Ergebnis	Testresultat
Benutzeroberfläche	Die Software wird gestartet	Die Benutzeroberfläche zeigt die aktuelle Uhrzeit minutengenau an	Testergebnis i.O.
	Die Software wird gestartet	Ein Logo soll auf der Benutzeroberfläche zu sehen sein	Testergebnis i.O.
Adminmenü	Es wird sich mit folgenden Nutzerdaten: ID: „123123“, Passwort: „654321“ angemeldet	Es soll das Adminmenü zu sehen sein	Testergebnis i.O.
	Es wird auf „Benutzerverwaltung“ geklickt	Hier sollen alle Benutzer angezeigt werden	Testergebnis i.O.
	Nun wird auf „Neuen Benutzer anlegen“ geklickt und ein Benutzer mit dem Namen: Rainer Zufall, Geburtsdatum: 05.04.1965, Wohnort: Fulda, Passwort: „123456“ und ID „100420“ angelegt	Es soll eine Fehlermeldung zu sehen sein, da es die ID schon gibt	Testergebnis i.O.
	Nun soll die ID „123456“ eingetragen werden	Das Menü soll sich schließen und es soll wieder die Benutzerverwaltung zu sehen sein	Testergebnis i.O.
	In der Benutzerliste wird nun der Name „Duebel, Arno“ ausgewählt und auf „Benutzer löschen“ geklickt	Es soll eine Infobox aufpoppen und mit dem Bestätigen dieser, soll der Benutzer nicht mehr in der Liste zu sehen sein	Testergebnis i.O.
	Der Benutzer „Schmidt, Stefan“ wird nun angeklickt. Anschließend wird auf „Benutzer bearbeiten“ geklickt. Der Benutzer wird nun den Namen „Steve Holzschuh“ bekommen und nun in Dortmund wohnen	In der Benutzerverwaltung soll dieser nun als „Holzschuh, Steve“ zu sehen sein	Testergebnis i.O.
	Danach wird in „Abwesenheiten“ durch einen Klick auf „Abwesenheit hinzufügen“ eine Abwesenheit für den Nutzer „Rainer Zufall“ erstellt. Dieser hat die ID: „123456“ und ist vom 12.07.2023 bis zum 14.07.2023 abwesend bzw. krank	In der Abwesenheitsliste soll der Name „Zufall, Rainer“ mit der ID: „123456“ und dem Abwesenheitszeitraum vom 12.07.2023 bis zum 14.07.2023 auftauchen	Testergebnis i.O.
	Danach wird auf den Reiter „Einstellungen“ geklickt und es wird folgendes eingestellt: Rundung der Arbeitszeit in Minuten: 5,	Nun sollen die neuen Einstellungen in der Einstellungs-Seite zu sehen sein	Testergebnis i.O.

	<p>Automatische Pausenzeit nach Stunden: 6</p> <p>Pfad des Firmenlogos: /data/Haftbefehl.png</p>		
	<p>Danach wird das Admin-Passwort geändert, indem wir auf „Neues Admin Passwort festlegen“ klicken. Hier geben wir nun das alte Passwort „123123“ ein und das neue Passwort „121212“</p>	<p>Wenn dies erfolgreich war, soll wieder die Einstellungs-Seite zu sehen sein</p>	<p>Testergebnis i.O.</p>
Init-Datei	<p>Nun wird die Init-Datei unter folgendem Pfad geöffnet: C:\Users\Samuel\source\repos\Working_time_management\Working_time_management\data\admin\init</p>	<p>Die init-Datei soll folgendermaßen aussehen:</p> <p>[Breaks] AfterHours=6</p> <p>[Time] Rounding=5</p> <p>[Logo] Path=/data/Haftbefehl.png</p>	<p>Testergebnis i.O.</p>
Benutzer	<p>Der Benutzer soll im Reiter „Zeiterfassung“ seine 6-stellige ID eingeben können und sich dann in der Arbeitssoftware als Anwesend eintragen</p>	<p>Der Benutzer gelangt ins Menü zur Zeiterfassung und sieht, nach einem Klick auf „Kommen/Gehen“, bei „Status“ in grüner Schrift „angemeldet“</p>	<p>Testergebnis i.O.</p>
	<p>Der Benutzer meldet sich mit der ID: „777888“ und dem Passwort: „987654“ an</p>	<p>Es soll eine Fehlermeldung aufpoppen, denn der Benutzer wurde gelöscht</p>	<p>Testergebnis i.O.</p>
	<p>Der Benutzer meldet sich nun mit der ID „123456“ und dem Passwort „123456“ an</p>	<p>Der Benutzer kommt ins Mitarbeitermenü</p>	<p>Testergebnis i.O.</p>
Benutzermenü	<p>Es wird im Menü auf Mitarbeiterinformationen, Urlaub und Überstunden geklickt</p>	<p>Der Benutzer soll seine Mitarbeiterinformationen einsehen können. Diese beinhalten Name, Resturlaub und Überstunden</p>	<p>Testergebnis i.O.</p>
	<p>Es wird im Reiter „Anträge“ ein neuer Antrag angelegt. Antragstyp: Urlaub, Von: 21.07.2023, Bis: 26.07.2023</p>	<p>Der Benutzer soll ohne Fehlermeldung zu der Antragsübersicht zurück gelangen. Im Admin-Menü unter dem Reiter Mitarbeiteranträge soll dieser Antrag nun aufgelistet sein</p>	<p>Testergebnis i.O.</p>
	<p>Es wird im noch ein Antrag angelegt, doch nun soll der Antragstyp ein</p>	<p>Der Benutzer soll ohne Fehlermeldung zu der Antragsübersicht zurück</p>	<p>Testergebnis i.O.</p>

	„Überstundenabbau“ sein, von: 17.07.2023, Bis: 19.07.2023	gelangen. Im Admin-Menü unter dem Reiter Mitarbeiteranträge soll dieser Antrag nun aufgelistet sein	
	Der Admin wird nun die beiden Anträge annehmen	Bei den Mitarbeiterinformationen sollen nun 26 verbleibende Urlaubstage und bei den Überstunden sollen 16 Stunden abgezogen werden. Bei den eben ertsellten Anträgen soll „genehmigt“ stehen	Testergebnis i.O.
CSV-Dateien und Ordnerstruktur	Die worker_information.csv unter folgendem Pfad: „C:\Users\Samuel\source\repos\Working_time_management\Working_time_management\data\worker_information\123456„ wird geöffnet	Die CSV-Datei sollte folgende Informationen enthalten: Zufall, Rainer, 04.05.1960, Fulda, Angemeldet	Testergebnis i.O.
	Im selben Ordner wird die „working_time.csv“ geöffnet	Dort sollte das aktuelle Datum mit der Uhrzeit der Anmeldung, der Uhrzeit der Abmeldung und der Zeitdifferenz, die gearbeitet wurde zu sehen sein	Testergebnis i.O.
	Danach wird die Datei „working_time_information.csv“ geöffnet	Hier soll bei Pause ein „n“, bei Übersunden soll -08:00 + die heute gearbeitete Zeit stehen und bei Resturlaub sollten 30 Tage stehen	Testergebnis i.O.
	Anschließend wird die Datei „request.csv“ geöffnet	Hier sollen die unter „Benutzermenü“ erstellten Anträge zu sehen sein	Testergebnis i.O.
	Danach wird zwei Ordner Ebenen nach oben gesprungen und auf den Ordner „id_pwd“ geklickt. Nun wird die darin befindliche CSV-Datei geöffnet	Hier sollen sämtliche IDs und Passwörter ablegen. Auch der Nutzer „Rainer Zufall“ mit der ID „123456“ und dem Passwort „123456“ soll hier aufgeführt werden	Testergebnis i.O.
	Nun wird wieder eine Ordner Ebene hochgesprungen und auf den Ordner „archive“ geklickt	Hier sollte der gelöschte Nutzer „Duebel, Arno“, ID: „777888“ mit seinen CSV-Dateien zu finden sein	Testergebnis i.O.
Arbeitszeit	Es wird sich noch einmal mit der ID: „123456“ in der Zeiterfassung	Es sollte bei „Status“ ein roter Schriftzug mit	Testergebnis i.O.

	angemeldet und sich als abwesend bzw. „abgemeldet“ eingetragen	„abgemeldet“ zu sehen sein	
	Nun öffnen wir wieder die CSV-Datei „working_time“	Es soll die Uhrzeit der Abmeldung vermerkt worden sein und die Arbeitszeit soll aus der Differenz zwischen An- und Abmeldung errechnet und gerundet worden sein	Testergebnis i.O.
	Nun wird die ID: „123456“ in der Zeiterfassung erneut angemeldet und nach 15 Minuten wieder abgemeldet	In der CSV-Datei „working_time“ soll nun ein weiterer Eintrag mit Kommen, Gehen und Arbeitszeit vorhanden sein	Testergebnis i.O.
	Nun wird der obere Testfall erneut ausgeführt, um zu zeigen, dass sich beliebig oft angemeldet werden kann	In der CSV-Datei „working_time“ soll nun ein weiterer Eintrag mit Kommen, Gehen und Arbeitszeit vorhanden sein	Testergebnis i.O.