
# Weiterentwicklung der labAlive-Webanwendung mit Schwerpunkt Websecurity

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Engineering (B.Eng.)

Ludwig Stöckl

Betreuer:
Prof. Dr.-Ing. Erwin Riederer

Tag der Abgabe: 30.06.2023

eingereicht bei
Universität der Bundeswehr München
Fakultät für Elektrotechnik und Technische Informatik

Neubiberg, Juni 2023

# Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen.

Der Speicherung meiner Bachelorarbeit zum Zweck der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

Neubiberg, den 30.06.2023

_____

Ludwig Stöckl

**Zusammenfassung**

Die labAlive-Webanwendung wird mit den Anforderungen, die bereits in der vorangegangen Projektarbeit definiert wurden, weiterentwickelt. Der Hauptteil der Arbeit konzentriert sich hierbei auf die Implementierung. Zum Schluss werden noch einige sicherheitstechnische Test und Beurteilungen durchgeführt, wobei der Schwerpunkt auf SQL-Injection und Cross-Site-Scripting Tests lag. Durch diese Weiterentwicklung wird labAlive zu einer sicheren und vertrauenswürdigen Plattform, die den Schutz der Benutzerdaten gewährleistet.

# Inhaltsverzeichnis

# 1 Einleitung

Die rasante Entwicklung der Technologie hat zu einer zunehmenden Digitalisierung in allen Bereichen unseres Lebens geführt, einschließlich der Art und Weise, wie wir mit Webanwendungen interagieren. In diesem Zusammenhang ist die Weiterentwicklung der labAlive-Webanwendung mit einem Schwerpunkt auf Websecurity von entscheidender Bedeutung. Websecurity bezieht sich auf die Implementierung von Maßnahmen, die die Vertraulichkeit, Integrität und Verfügbarkeit von Webanwendungen gewährleisten, um potenzielle Angriffe zu verhindern und sensible Informationen zu schützen. Angesichts der zunehmenden Bedrohungen durch Cyberkriminalität ist es unerlässlich, dass labAlive kontinuierlich verbessert wird, um den Anforderungen der modernen Websecurity gerecht zu werden. Im Folgenden wird näher auf die Bedeutung dieser Weiterentwicklung eingegangen und die Vorteile für die Benutzer der labAlive-Webanwendung erläutert.

## 1.1 Aufgabenstellung

Das Ziel dieser Arbeit ist es, das bereits definierte System zu implementieren. Das neue System soll effizienter bei Zugriffen auf die einzelnen Daten sein und soll diese vor unerlaubten Zugriffen schützen. Die Webanwendung soll auf alle vorhanden Ressourcen, welche labAlive zu bieten hat, erweitert werden und den vollen Funktionsumfang der einzelnen Resourcen zur Verfügung stellen. Ein wichtiger Gesichtspunkt hierbei ist die Sicherheit der Daten, welche durch Angriffe ausspioniert werden könnten. Um dies zu verhindern, beschäftigt sich die Arbeit zum Schluss mit den 2 wichtigsten Sicherheitslücken im Web, um unerlaubte Zugriffe zu verhindern und die Webanwendung sicherer zu machen.

# 2 Software und Programmiersprachen

## 2.1 Software

Jede Entwicklungsumgebung bringt Vorteile und Nachteile mit sich. Deshalb war es an vielen Stellen produktiver, die einzelnen Teile mit verschiedenen Programmiersprachen bzw. Software zu entwickeln und am Schluss zusammenzuführen.

### 2.1.1 Microsoft Visio

Microsoft Visio ist ein Visualisierungsprogramm von Microsoft für Windows und wurde ursprünglich von der Firma "Visio Corporation"1992 entwickelt. Im Januar 2000 wurde das Unternehmen von Microsoft für 1,3 Milliarden US-Dollar aufgekauft. In Visio können Schaubilder und Diagramme mit Hilfe vielfältiger unterschiedlichen Vorlagen und Werkzeugen sehr effektiv erstellt werden. Das Programm ist besonders gut geeignet Ablaufdiagramme, Organigramme und Geschäftsprozesse graphisch darzustellen. Aber auch einfache technische Zeichnung und UML-Diagramme können mit dem Programm erstellt werden. Der Vorteil hierbei ist, dass einzelne Shapes mit beliebigen Datenbanken und Excel-Tabellen verknüpft werden können. [1]

### 2.1.2 Gitea

Gitea ist eine freie Softwareentwicklunsplattform, die in Go entwickelt wurde. Die Plattform stellt eine Versionsverwaltung über Git zur Verfügung. Sie enthält darüber hinaus jedoch weitere kollaborative Werkzeuge wie z.B. einen Bugtracker, ein Wiki und ein Code Review Modul. Die Benutzeroberfläche orientiert sich an der Oberfläche von GitHub. Das System kann auf verschiedenen Systemen installiert werden. Aufgrund der geringen Hardwarevoraussetzungen kann die Plattform im Gegensatz zu GitHub auch auf Embedded Systemen installiert werden. [2]

### 2.1.3 IntelliJ

IntelliJ ist eine Entwicklungsumgebung von JetBrains, welche speziell für die Programmiersprachen Java, Kotlin, Groovy und Scala entwickelt worden ist. Ab der Version 9.0 gibt es zwei verschiedene Editionen, eine kostenpflichtige „Ultimate Edition“ und eine kostenlose „Community Edition“. Der größte Unterschied zwischen der Community und Ultimate Edition ist die Unterstützung von mehreren Programmiersprachen. Die Community Edition unterstützt nicht so viele Sprachen unteranderem JavaScript um ein Beispiel zu nennen. IntelliJ unterstützt weitere Programmiersprachen wie zum Beispiel Java EE, Apache Maven Tools zur Versionskontrolle, insbesondere Git, automatisches Refactoring von Code und einen Decompiler für

Java-Klassen. Die IDE kann mittels Plugins erweitert werden, welche von der Community und JetBrains entwickelt werden. Die Firma unterstützt besonders die Entwicklung der Plugins von der Community. Hierfür organisiert die Firma jedes Jahr einen Wettbewerb um die Weiterentwicklung zu unterstützten. Für diese Arbeit wurde die Ultimate Edition ausgewählt, da diese kostenlos für Studenten zu Verfügung steht und da diese Programmiersprachen zu Verfügung stellt, die für dieses Projekt benötigt werden und nicht in der Community Edition nicht zur Verfügung stehen. [3]

### 2.1.4 Microsoft Access

Microsoft Access ist eine Datenbank Anwendung, welche über die Microsoft-Office-Familie (Microsoft 365) erworben werden kann. Die Software kombiniert ein relationales Datenbankmanagementsystem (Microsoft Jet Engine) mit den Werkzeugen einer integrierten Entwicklungsumgebung. MS Access unterstützt auch die Datenbank-Programmiersprache SQL und eignet sich besonders mit seiner grafischen Benutzeroberfläche für die Zielgruppe der Endbenutzer. [4]

### 2.1.5 Burp Suite Community Edition

Burp Suite ist eine Java-Anwendung der Firma PrortSwigger deren Hauptanwendung das Testen und Analysieren der Sicherheit von Webanwendungen ist. Das Programm umfasst einen Proxy-Server, einen Spider, einen Intruder und einen sogenannten Repeater für die Automatisierung von Anfragen. Die Anwendung ist mittlerweile der Industriestandard für die Durchführung von Penetrationstests. Von Burp Suite gibt es mehrere kostenpflichtige Versionen und eine kostenlose Version, die Community Edition. [5]

### 2.1.6 OWASP Zed Attack Proxy (ZAP)

Diese Anwendung ist ein Open-Source-Sicherheitsscanner für Webanwendungen. Die Anwendung ist sowohl für Neueinsteiger in dem Bereich Anwendungssicherheit also auch für professionelle Penetrationstester geeignet. Die Software ist ein Projekt des Open Web Application Security Project (kurz: OWASP), welches den Flagship-Status erhalten hat. ZAP kann als Proxyserver verwendet werden und ermöglicht somit dem Benutzer die Manipulation des gesamten Datenverkehrs. Das Programm kann aber auch im Damon Modus ausgeführt werden, wobei es in diesem Modus über eine REST-API gesteuert wird. [**ZAP**]

## 2.2 Programmiersprachen

Zur Weiterentwicklung der labAlive Webanwendung kamen unterschiedliche Programmiersprachen zum Einsatz, welche im Folgenden aufgelistet sind.

### 2.2.1 Java

Java ist eine objektorientierte Programmiersprache, welche hauptsächlich zum Formulieren von Programmen dient. Java ist auch eine eingetragene Marke des Unternehmens Sun Microsystems das von Oracel aufgekauft wurde. [6]

### 2.2.2 JavaScript

JavaScript ist eine Skriptsprache, die für dynamisches HTML im Webbrowser entwickelt wurde. Die Sprache wurde 1995 ursprünglich von Netscape entwickelt und wird verwendet um Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren. HTML wird damit um diese Funktionen erweitert. Darüberhinaus wird derzeit JavaScript auch außerhalb von Browsern angewendet, z.B. auf Server oder in Microcontrollern. [7]

### 2.2.3 HTML

HTML ist eine textbasierte Programmiersprache, die zum Strukturieren elektronischer Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten dient. Sie bildet die Grundlage des World Wide Web und wird von den Webbrowsern verwendet. HTML Code enthält häufig zusätzliche Angaben wie Metainformationen, die Angaben über die im Text verwendete Sprache oder den Autor enthalten. HTML dient dazu, einen Text semantisch zu strukturieren, ihn jedoch nicht zu formatieren. [8]

### 2.2.4 SQL

SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten und Abfragen von Datenbeständen. Die Sprache basiert auf der relationalen Algebra. Die Syntax ist relativ einfach aufgebaut und semantisch an die englische Umgangssprache angelehnt. [9]

# 3 Überblick

Die Grundlage dieser Bachelorarbeit ist die vorausgegangene Projektarbeit mit dem Titel "Bestandsaufnahme, Codeverbesserung und Anforderungen der labAlive-Webanwendung". In der Projektarbeit wurden die Anforderungen an die Webanwendung definiert, welche dann im Zuge dieser Bachelorarbeit implementiert und auf ihre It-Sicherheit untersucht wurden. Im Folgenden wird die Projektarbeit, die als Grundlage für diese Bachelorarbeit diente, kurz erläutert.

## 3.1 labAlive

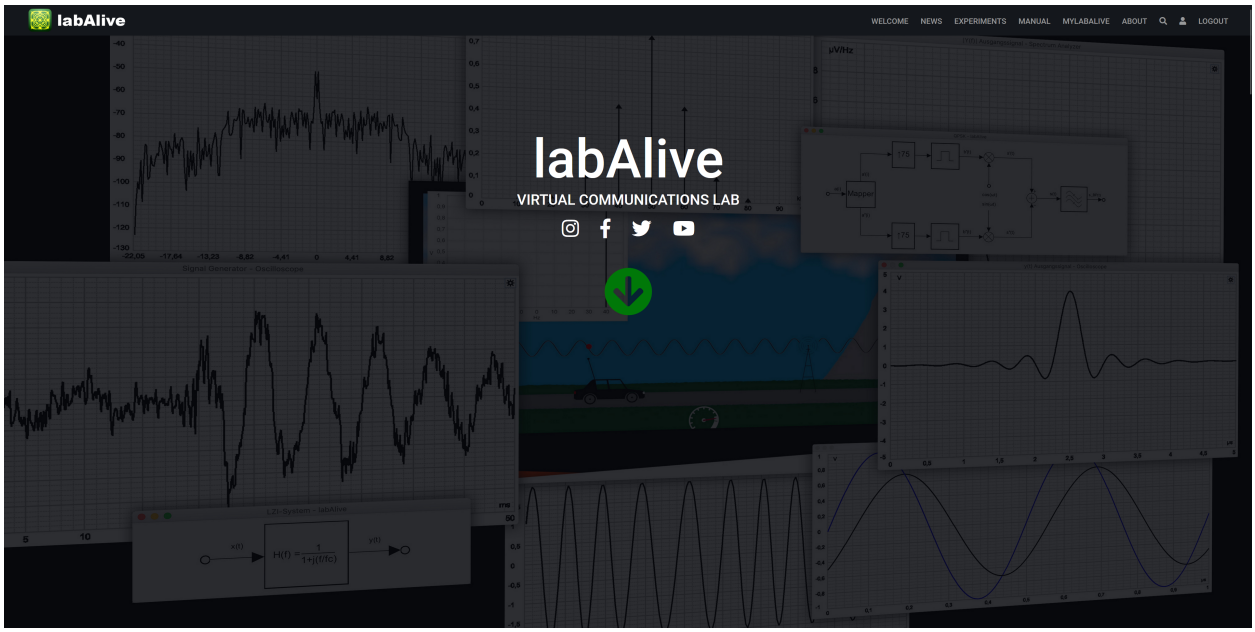


Abbildung 3.1: labAlive-Startseite

LabAlive stellt eine digitale Experimentierumgebung für verschiedene Übertragungsverfahren, Nachrichtensysteme sowie Signalmodulationen bereit. Es können mithilfe von digital modellierten Messgeräten Signale dargestellt, simuliert und auch verändert werden. Der aktuelle Stand der Webanwendung beruht darauf, dass aktuell nur ein sogenanntes ’Wiring’ umgesetzt werden kann. Das ’Wiring’ ist im Prinzip die Grundlage für die digitale Simulation. Es beinhaltet die textbasierten Formeln um ein Signal z.B. eine Nachrichtensignal zu erzeugen. Um den vollen Funktionsumfang der Webanwendung zu nutzen, wurden in der Projektarbeit die Anforderungen definiert, welche im Folgenden vorgestellt werden.

## 3.2 Datenbankmodell

Als Erstes wird das Datenbankmodell vorgestellt, welches auch an erster Stelle der Implementierung stand. Hierzu wurde auch ein Objektmodell entwickelt, welches sich auch im dem Datenbankmodell widerspiegelt. Übergeordnet steht das Grundgerüst die "Resource". Sie bildet die Basis und speichert die grundlegenden Informationen, wie UserID, Titel, Description, etc. Eine wichtig Variable ist unter anderem auch ist der Published Status. Er gibt an, ob eine Resource veröffentlicht ist bzw. auf 'privat' gestellt ist. Ein weiteres wichtiges Attribut dieses Objektes ist der "Type". Dieser enthält die Information welcher Typ geladen wird. Es werden hierbei 7 verschiedene Typen unterschieden: Wiring, Signal, RunWiring, QueryString, Laxout, Image, und Experiment. Diese sind auf der rechten Seite in der Abbildung 3.1 dargestellt. Diese Typen erben von dem Objekt Resource. Somit besteht eine komplette Resource aus einem Typen (Bsp.: Wiring) und der Grundstruktur "Resource". Es gibt noch zusätzliche Objekte die zu der Grundstruktur 'Resource' gehören. Das sind dann die "Keywords", "Formula", Image","LinkedResources". "Keywords" sind Schlagwörter, welche dazu dienen die Resourcen zu sortieren zu filtern bzw. um diese auch zu suchen. Es können jeder Resource mehrere Keywords zugewiesen werden. "Formula" sind, wie der Name schon erahnen lasst, bestimmte Formeln, welche den Resourcen zugeordnet werden können, genauso wie „Image". Die "LinkedResources" sind etwas besonders, denn eine Resource kann auf mehrere andere Resourcen verlinken. Da es jedoch bei einer Resource mehrere Felder bzw. Typen zum verlinken gibt (Bsp.: Created With, Used With, etc) ergab sich hier ein Problem. Da aufgrund der relationalen Datenbank keine Listen in eine Zelle gespeichert werden können, wurden für dieses Projekt einfach mehrere Tabellen Einträge angelegt. Vorgehen und der Ablauf wird im nächsten Unterkapitel erklärt.

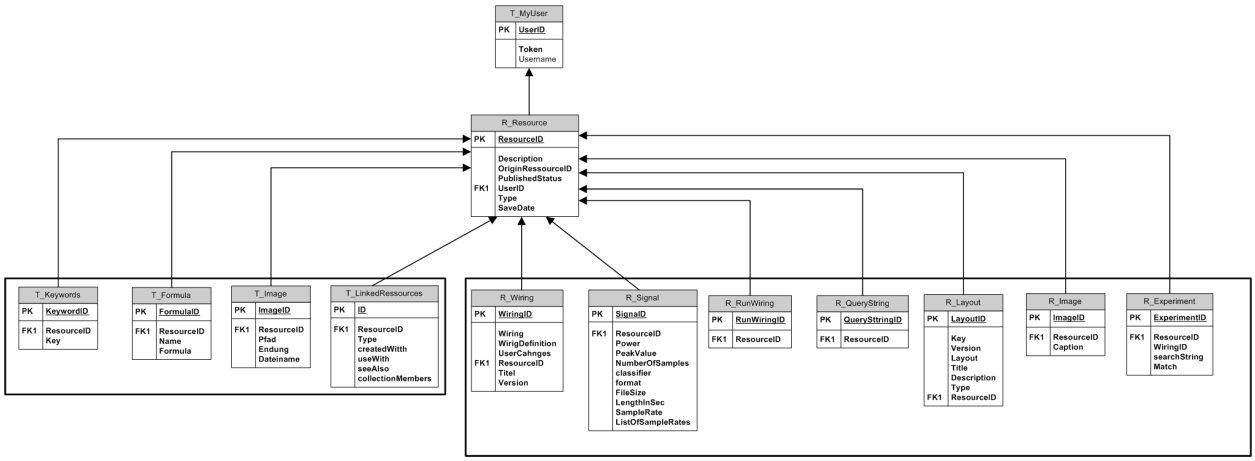


Abbildung 3.2: Datenbank-Diagramm

## 3.3 Ablaufdiagramme

In diesem Unterkapitel werden die erstellten Ablaufdiagramme dargestellt und erläutert, die ebenfalls als Grundlage für die Bachelorarbeit dienten.

### 3.3.1 Anlegen einer Resource

In diesem Kapitel wird das Anlegen und Speichern einer Resource beschrieben. Wenn ein Benutzer zum Beispiel ein Wiring anlegt und die Benutzereingaben in der Anwendung speichert, dann wird zuerst eine normale Resource angelegt und vorab mit den nötigen Metadaten befüllt. Zuerst wird die Resource in der Datenbank angelegt und gespeichert um dann als nächstes dem Cache hinzugefügt. Da der Benutzer gleich eine spezalisierte Resource speichert, wird der Resourcen-Typ zu der Resource hinzugefügt. Somit kann diese dann auch im Cache gespeichert werden und der richtigen Liste zugeordnet werden. Zum Schluss werden noch die spezialisierten Informationen gespeichert, in diesem Beispiel die Daten, die für die WiringResource benötigt werden. Diese werden in die Datenbank geschrieben und gespeichert. Da die Resource den Listen im Cache schon hinzugefugt worden sind, kann auf ein separates Update des Cache verzichtet werden.

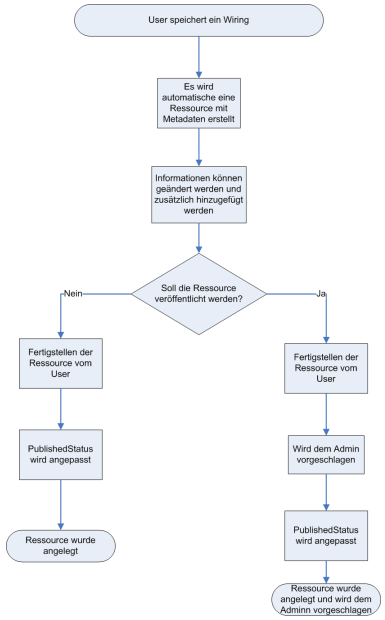


Abbildung 3.3: Ablaufdiagramm Wiring speichern

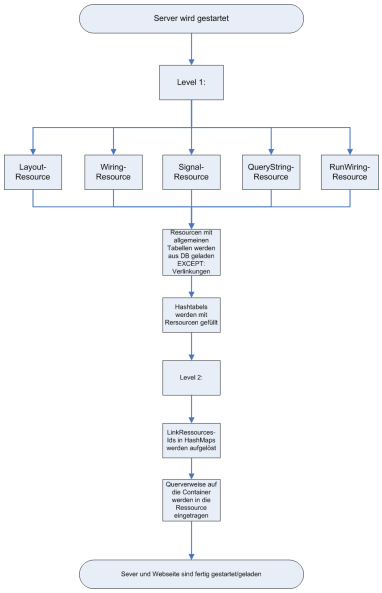


Abbildung 3.4: Ablaufdiagramm Server-start

### 3.3.2 Server-Start

Abbildung 3.3 zeigt das Ablaufdiagramm, zum Start des Servers und zum Laden einer Resource in den Cache. Nachdem Server-Start gibt es 2 Level in die eine komplette Resource geladen wird. Im Level-1 wird zuerst unterschieden welche spezialisierte Form die Resource hat, um dann diese in die entsprechenden Hashtabels des Cache zu laden. Dabei werden auch die ganzen spezialisierten Daten der Resource, sowie deren allgemeinen Informationen geladen ausgenommen hiervon sind die LinkedResources. Bei den LinkedResources, welche in Level-2 geladen werden, werden zuerst nur Querverweise in die Resource geladen und später beim Anzeigen die restlichen Informationen. Wenn nun eine Resource angezeigt werden soll, wird diese direkt aus dem Cache geladen und angezeigt.

# 4 Implementierung

Nun zur eigentlichen Bachelorarbeit. Diese bestand aus der Implementierung der Applikation und der dazu gehörigen Web-Security. Als Erstes wurde die Implementierung angegangen. In diesem Projekt gibt es mehrere Layer: den Datanbank Layer, den Objekt/Data Layer, den Cache/Provider Layer, den Servlet Layer und den JSP Layer.
Herr Dombek und ich haben zusammen die Objekte und Servlets umgesetzt, wobei der Fokus von Herr Dombek auf dem Cache, während mein Fokus auf der Datenbankseite lag. Der erste Layer ist der Datenbank-Layer, welcher die Datenbank, sowohl aber auch die Klassen zum Herauslesen der Daten enthält. Beim Herauslesen der Daten werden selbige gleich in die angelegten Objekte geschrieben. Als Nächstes werden die einzelnen Objekte in den Cache geschrieben und in Listen einsortiert. Zum Schluss werden diese aus dem Cache in die JSP geladen und auf der Webseite angezeigt.

Um nun in der Datenbank eine Resource anzulegen, zu aktualisieren oder zu löschen wird ein Servlet aufgerufen, welches die Parameter von der Webseite (dem JSP Layer) entgegennimmt und diese an den Provider/Cache Layer weiterleitet. Dort werden die Objekte entsprechend geändert und gleichzeitig auch in die Datenbank zurückgeschrieben. Da die Objekte in dem Cache bereits geändert werden, erfordert dies keine separate Änderung.

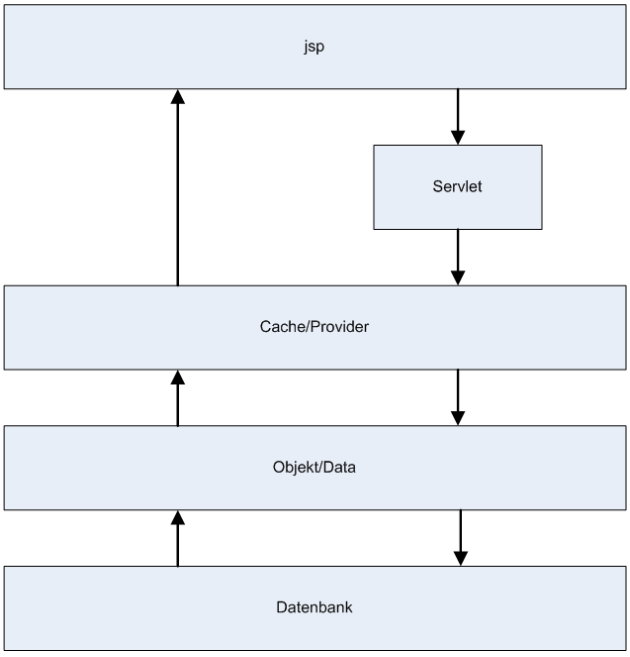


Abbildung 4.1: Layer-Diagramm

## 4.1 Objekte/Data

Als Erstes wurden die Objekte umgesetzt. Um nicht ständig die Listen im Cache anzufassen und neu zu schreiben, wurde die Resource in einen Resource Container gepackt. Dadurch kann der Resource Container weiterhin im Cache stehen nur die Resource, die im Container liegt, muss ausgetauscht bzw. zu aktualisiert werden. Um dieses zu realisieren wurden als erstes die Resource-Objekte angelegt. Das Objekt Resource-Data wurde zuerst erstellt und es wurden alle Attributte des Objektes, zusammen mit den ganzen 'Getter' und 'Setter' Methoden, angelegt. Das sind spezielle Methoden eines Objektes, die ein einzelnes Attribut eines Objektes abfrägt oder ändert. Somit sind diese Methoden Teil der öffentlichen Schnittstelle des Objektes und verbergen mehrere Implementierungsdetails. Mit Hilfe dieser Methoden kann die Implementierung eines Objekts geändert werden, ohne seine öffentliche Schnittstelle zu ändern. Die 'Getter'-Methode ist eine Zugriffsmethode, welche eine Eigenschaft eines Objektes abfrägt. Durch diese Möglichkeit kann die Eigenschaft direkt aus einem Objektattribut entnommen werden. Bei der Setter-Methode kann nun ein Objekt geändert werden. Der Vorteil dabei ist, dass bei der Änderung der Wert auf Gültigkeit geprüft werden kann. Wenn zum Beispiel ein ungültiger Wert übergeben wird, kann als eine mögliche Reaktion eine Ausnahmeroutine ausgelöst werden, wodurch der normale Programmablauf unterbrochen und ein Fehler signalisiert wird.

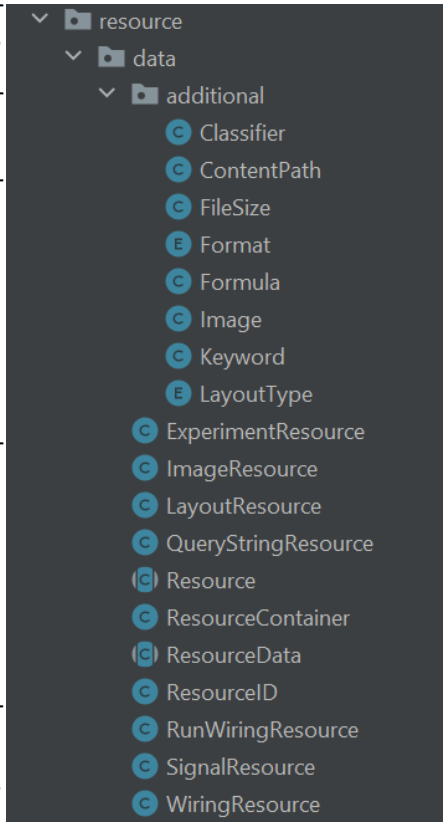

Abbildung 4.2: Resourcenübersicht

Da dieses Objekt als Grundlage für die weiteren Objekte dient, wurde diese Klasse als abstrakt deklariert. Als 'Abstrakte Klasse' wird eine Klasse bezeichnet, von der keine Objekte erzeugt werden können und sind somit nicht 'vollständig' genug, um Objekte zu instanziieren. Als Nächstes wurde die Klasse Resource erstellt, welche von der abstrakten Klasse ResourceData erbt. Eine Vererbung dient dazu neue Klassen auf Basis der alten Klasse zu schaffen, wobei die neue Klasse eine Erweiterung oder eine Einschränkung sein kann. Jedoch ist die Beziehung zwischen den beiden Klassen dauerhaft. Da nun Resource-Data die Basisklasse ist, ist Resource nun die abgeleitete Klasse aus der Basisklasse. Das bedeutet, dass nun die erbenden Klassen, die Ableitung bzw. die Spezialisierung der Resource-Data Klasse sind. In der Resource stehen jedoch nur dedizierte Links für bestimmte Anwendungen. Danach wurde der ResourceContainer erstellt, welcher nur aus dem Attribut Resource besteht. Neben der 'Getter' und 'Setter-Methode' für die Resource wurde noch eine 'createMethode' implementiert. Mit dieser Methode, die eine Resource übergibt, wird eine neue Instanz des ResourceContainer für die übergebene Resource gesetzt. Nachdem das nun erledigt war, konnten die Spezialisierung der Klasse Resource in Angriff genommen werden.

```
11      private long resourceID;
        2 usages
12      private String titel;
        2 usages
13      private String description;
        2 usages
14      private long originResourceID;
        2 usages
15      private int publishedStatus;
        2 usages
16      private long userID;
        2 usages
17      private String resourceType;
        2 usages
18      private String saveDate;
19
        1 override  Niklas Dombek +1
20      public long getResourceID() { return resourceID; }
23
        1 override  Niklas Dombek +1
24      public void setResourceID(long resourceID) { this.resourceID = resourceID; }
27
        l_stoeckl
28      public String getTitel() {
29          return titel;
30      }
31
        l_stoeckl
32      public void setTitel(String titel) {
33          this.titel = titel;
34      }
35
        1 override  riederer
36      public String getDescription() { return description; }
39
        1 override  riederer
40      public void setDescription(String description) { this.description = description; }
43
```

Abbildung 4.3: Resource

Dazu wurde analog zu ResourceData die Klassen ExperimentResource, ImageResource, LayoutResource, QueryStringResource, RunWiringResource, SignalResource und WiringResource angelegt. Jedoch sind diese Klassen nicht abstrakt, da von den Klassen Instanzen erzeugt werden müssen. Auch wurde eine 'createMethode' wie bei dem ResourceContainer implementiert, jedoch wird hier nur eine ID übergeben und nicht ein komplettes Objekt. Somit sind nun alle Objekte angelegt und als Nächstes wurde sich dem Provider und der Datenbank gewidmet.

## 4.2 Datenbank

Da Herr Dombek sich um den Provider bzw. den Cache kümmerte, oblag mir im Rahmen dieser Bachelorarbeit die Implementierung der Datenbank-Anbindung. Bedingt durch die Tatsache dass die Webseite schon seit geraumer Zeit besteht, bestand schon die Funktionalität der Verbindung zur Datenbank. Was bereits umgesetzt war, sind die Klassen DBCommand, DBConnectionManager, DBSelectCommand, DBUpdateCommand. Die Aufgabe bestand nun darin die definierten Klassen zu nutzen, um die Daten aus den richtigen Tabellen abzufragen und die jeweiligen Objekte zu laden. Dazu wurde zuerst jeweils eine Datanbank-Klasse zu jedem Objekt angelegt, da jedes Objekt seine eigene Tabelle in der Datenbank besitzt. Begonnen wurde mit dem Adapter für die Resource, da dieser geringfügig anders aufgebaut ist als die Spezialisierungen. Hierbei wurden die Aktionen in mehrere Methoden aufgeteilt, da mehrere 'Commands' erstellt werden mussten. Folgende 4 Arten von Aktionen gibt es: get, create, update und delete. Begonnen wurde mit der get-Aktion, welche in 2 verschiedene Methoden aufteilt wurde und zwar zwischen dem Erstellen des Commands und dem Ausführen des Commands. Am Anfang wurde die Methode 'getAllResources' gesetzt, welche den Command ausführt und eine Collection an Resourcen zurückgibt. In dieser Methode wurde auch die Methode zum Erstellen des Commands aufgerufen.

```
26      public static Collection<Resource> getAllResources() throws DBException {
27          DBSelectCommand cmd = getAllResourcesCmd();
28          return (Collection<Resource>) cmd.run();
29      }
```

Abbildung 4.4: getAllResource-Methode

Diese Methode wurde gleich als Nächstes erstellt, in welcher zunächst ein String erstellt wurde, der einen SQL-Befehl darstellt. Um Informationen aus den Tabellen einer Datenbank abzufragen, gibt es den 'SELECT'-Befehl. Der SQL-Befehl ist folgendermaßen aufgebaut: 'SELECT * FROM Resource'. Zuerst kommt der 'SELECT'-Befehl, danach kommt der Spaltenname, welcher ausgewählt werden soll, danach kommt das Schlüsselwort 'FROM' und anschließend der Tabellenname in der die Spalten existieren müssen, die ausgewählt wurden. Sind diese über 'SELECT' ausgewählten Spalten nicht in der angegebenen Tabelle, wird ein leeres Ergebnis zurückgegeben. Generell wird durch den 'SELECT'-Befehl eine virtuelle Tabelle erzeugt, die anschließend ausgegeben wird. Falls ein Stern anstelle des Spaltennamen angegebenen wird, wird die komplette Tabelle zurückgegeben. Als Nächstes wurde eine neue Instanz der DBSelectCommand-Klasse erstellt, welcher den String übergeben wurde. Um nun die Ergebnisse der Abfrage der Datenbank gleich abzufangen und abzuspeichern wurde eine kleine Untermethode namens 'handleResultSet' geschrieben. In dieser wurde zuerst eine neue ArrayListe mit dem Typen 'Resource' angelegt, da hier mehrere Ergebnisse zu erwarten sind. Als Nächstes wurde eine 'whileSchleife' erstellt, die über die Ergebnisstabelle iteriert

```
31  @   private static DBSelectCommand getAllResourcesCmd() throws DBException {
32          String preparedStat = "SELECT * FROM R_Resource";
            Niklas Dombek +3
33          DBSelectCommand cmd = new DBSelectCommand(preparedStat) {
                Niklas Dombek +3
34  @           protected Object handleResultSet(ResultSet rs) throws Exception {
35              Collection<Resource> resources = new ArrayList<>();
36              while (rs.next()) {
37                  Resource resource = createReturnResourceType(rs.getString( columnLabel: "ResourceType"));
38                  resource.setResourceID(rs.getLong( columnLabel: "ResourceID"));
39                  resource.setTitel(rs.getString( columnLabel: "Titel"));
40                  resource.setDescription(rs.getString( columnLabel: "Description"));
41                  resource.setOriginResourceID(rs.getLong( columnLabel: "OriginResourceID"));
42                  resource.setPublishedStatus(rs.getInt( columnLabel: "PublishedStatus"));
43                  resource.setUserID(rs.getLong( columnLabel: "UserID"));
44                  resource.setResourceType(rs.getString( columnLabel: "ResourceType"));
45                  resource.setSaveDate(rs.getString( columnLabel: "SaveDate"));
46                  resources.add(resource);
47              }
48
49              return resources;
50          }
```

Abbildung 4.5: getAllResourceCmd - Teil 1

Durch das 'rs.next' wird jeweils eine Zeile weiter gegangen und zwar durch alle Zeilen bis es keine Zeilen mehr gibt. Um nun die richtigen Spalten und die richtigen Werte den richtigen Attributen des Objektes zuzuordnen, wird mit Hilfe des Spaltennamens die 'Strings' geholt, und diese gegebenenfalls in die richtigen Datentypen umgewandelt und diese direkt in das Objekt Resource geschrieben.

Abbildung 4.6: getAllResourceCmd - Teil 2

In der Untermethode wird auch eine Resource anhand des Resourcetypes, welcher in der Resource gespeichert ist, erstellt. In diese Resource werden nun die Daten geschrieben und zu der bereits erstellten Liste hinzugefügt. Dadurch, dass die Resource die 'Elternklasse' ist, kann diese auch später als Kindklasse definiert werden. Mit Hilfe der switch-case Anweisung kann nun durch den Resource-Typen die Spezialisierung ausgewählt werden. Bei dieser Verzweigung können Werte überprüft werden und dadurch dann auch selektive Anweisungen ausgeführt werden. Je nachdem welche Verzweigung nun gewählt wird, wird die spezialisierte Resource zurückgegeben. Zum Schluss der Methode wird der Command zurück gegeben.

Der nächste Schritt war die Erstellung der Methode, mit welcher in der Datenbank ein neuer Eintrag erstellt werden kann. Der Methode wird, wie in der Abbildung gezeigt, eine Resource übergeben. Die erste Aktion, die in dieser Methode ausgeführt wird, ist das Speichern der Objekt-Attribute, welche von der Webseite kommen, in die separaten Variablen. Als Nächstes wird der String für die SQL-Abfrage vorbereitet, dabei wird das Schlüsselwort 'INSERT INTO' verwendet. Das bewirkt, dass Daten in die Tabelle geschrieben werden, welche direkt hinter diesem Schlüsselwort steht. Danach kommen nun die Spaltennamen, in die die Daten geschrieben werden sollen. Um hier die Lesbarkeit zu erhöhen, wurden

die Spaltennamen extra eingeklammert.

```java
public static Resource createResource(Resource resource) throws DBException {
    String description = resource.getDescription();
    String titel = resource.getTitel();
    long originResourceID = resource.getOriginResourceID();
    int publishedStatus = resource.getPublishedStatus();
    long userID = resource.getUserID();
    String resourceType = resource.getResourceType();
    String saveDate = resource.getSaveDate();
    String uuid = UUID.randomUUID().toString();

    String preparedStat = "INSERT INTO R_Resource (Titel, SaveDate, Description, UserID, OriginResourceID, PublishedStatus, ResourceType, UUID ) VALUES ( "
            + "? , "//Titel
            + "? , "//Description
            + "? , "//OriginResourceID
            + "? , "//PublishedStatus
            + "? , "//UserID
            + "? , "//ResourceType
            + "? , "//SaveDate
            + "? )";//UUID
    Object[] parameterList = {titel, saveDate, description, userID, originResourceID, publishedStatus, resourceType, uuid};
    DBUpdateCommand cmd = new DBUpdateCommand(preparedStat, parameterList);
    System.out.println(preparedStat);
    cmd.run();

    long resourceID = 1;
    try{
        resourceID = getResourceIdByUUID(uuid); //gets current resourceId
    } catch (Exception e){
        e.printStackTrace();
    }
    resource.setResourceID(resourceID); //Sets proper resourceId
    return resource;
}
```

Abbildung 4.7: create-Methode

Nach dieser Klammer kommt das Stichwort 'VALUES' welches signalisiert, dass die Werte für die Spalten folgen. Um nun einer SQL-Injection vorzubeugen, wurden in der Klammer nur Fragezeichen eingetragen, die wieder eingeklammert wurden. Wieso jetzt genau nur Fragezeichen verwendet wurden, wird im Kapitel 'Websecurity - SQL-Injection' genauer erklärt. In der folgenden Variable werden als Nächstes die Parameter in ein Array geschrieben, welche zusammen mit dem SQL-Statement der neuen Instanz von der Klasse 'DBUpdateCommand' übergeben wird. Mit dem Befehl 'cmd.run()' wird jetzt der SQL-Befehl ausgeführt und die Daten in die Datenbank geschrieben. Um nun die ID, welche von der Datenbank automatisch vergeben wird, zu bekommen, wurde eine UUID angelegt, die auch in die Datenbank geschrieben wurde. Da die UUID eine eindeutig generierte Zahl ist, kann mit Hilfe dieser die ID aus der Datenbank für den soeben gespeicherten Datensatz heraus gelesen werden. Dies geschieht hier wieder in einer neuen Methode, welche auch mit dem 'SELECT'-Statement passiert. Die Methode gibt nun speziell die ID als eine Long-Variable zurück. Diese wird nun nochmals dem übergebenen Objekt hinzugefügt und dieses Objekt wird derweilen auch wieder zurückgegeben.

```java
public static void updateResource(Resource resource) throws DBException {
    String preparedStat = "UPDATE R_Resource SET Titel = ?, SaveDate = ?, Description = ?, PublishedStatus = ?, ResourceType = ? WHERE ResourceID = ?";
    Object[] parameterList = {
            resource.getTitel(),
            resource.getSaveDate(),
            resource.getDescription(),
            resource.getPublishedStatus(),
            resource.getResourceType(),
            resource.getResourceID()};
    DBUpdateCommand cmd = new DBUpdateCommand(preparedStat, parameterList);
    cmd.run();
}
```

Abbildung 4.8: update-Methode

Jetzt fehlen nur noch 2 Methoden, die 'Update'- und 'Delete'-Methode. Diese beiden Methoden sind relativ kurz. Zuerst zu der 'Update'-Methode. Bei dieser wird zuerst wieder das SQL-Statement definiert und zwar diesmal mit dem Wort 'UPDATE'. Nachdem Wort kommt wie immer die Tabelle, in der der Datensatz aktualisiert werden soll. Danach kommt das Wort 'SET' und danach jeweils der Spaltenname mit dem Wert. Zum Schluss kommt 'WHETE' und danach wieder ein Spaltenname mit dem dazu gehörenden Wert. Diese Syntax ist dafür verantwortlich, dass die Zeile im Datensatz an der richtigen Stelle aktualisiert wird. Nach dem Definieren das SQL-Statement wird eine Liste an Parametern angelegt, welche wieder zusammen mit dem Statement zusammen übergeben wird. Zum Schluss wird der Befehl ausgeführt.
Zum Schluss der Klasse wurde noch die 'Delete'-Mehtode umgesetzt. Diese Methode ist genauso wie die Update-Methode aufgebaut. Jedoch statt 'UPDATE' wird 'DELETE' verwendet und der 'SET'-Teil wird nicht verwendet. Mit Hilfe der ID wird der Datensatz gesucht und dieser dann gelöscht. Die Datenbank-Adapter für die Spezialisierungen der Resource wurden analog zu diesem Adapter angelegt und somit ist die Datenbank nun mit den Objekten verbunden.

## 4.3 Servlet

Als Nächstes wurden nun die Servlets und die Klassen, die vor den Servlets sind in Angriff genommen. Hier wurde die Arbeit auch zwischen Herrn Dombek und mir aufgeteilt. Jeder programmierte die Hälfte. Bevor die Servlets anlegten werden konnten, wurden noch Zwischen-Klassen angelegt. Diese sind dafür da die Daten von den Servlets an den Provider und der Datenbank weiterzuleiten. Dazu wurden 2 verschiedene Arten von Klassen angelegt, wie zum Beispiel 'WiringDB4Servlet' und 'WiringSelector4Jsp'. Anhand dieser beiden Beispiele wird soll nun die Funktion der Klassen vorgesellt werden. Die Klasse DB4Servlet ist dafür da sowohl den Cache als auch die Datenbank zu aktualisieren, ob eine sofern im Fall WiringResource erstellt, aktualisiert oder gelöscht wird.

```
26 @    public static WiringResource createWiring(WiringResource wiring) throws DBException {
27          wiring = (WiringResource) ResourceDBInternal.createResource(wiring); // resourceId is set
28          wiring = R_wiringDBImpl.createWiring(wiring);   // version is set / might be changed
29          return wiring;
30      }
```

Abbildung 4.9: WiringDB4Servlet-createMethode

Wie in der Abbildung gezeigt, wird zuerst der Cache aufgerufen und dort zuerst aktualisiert und danach der Datenbank-Adapter. Dabei wird jeweils die Resource übergeben. Die anderen Methoden sind analog aufgebaut. Nun gibt es noch die 'Selector4Jsp'-Dateien. In diesen Dateien werden nur die Collections von spezialisierten Resourcen zurückgegeben, damit diese auf der JSP angezeigt werden können. Nachdem nun diese Klassen existieren, konnten die Servlets angelegt werden. Als Servlet bezeichnet man Java-Klassen, die Anfragen von Clients innerhalb von Webservern entgegennehmen und beantworten. Der Inhalt dieser Anfragen kann dabei dynamisch, aber auch statisch sein. Die Servlets in dieser Anwendung erben von 2 Servlet-Klassen, welche bereits von der Java-Bibliothek zur Verfügung gestellt werden.

```
18  @WebServlet(urlPatterns = { "/wiring/createupdate/" })
19  public class WiringServlet extends javax.servlet.http.HttpServlet implements javax.servlet.Servlet{
20
```

Abbildung 4.10: WiringServlet

Wie aus der obigen Abbildung ersichtlich, wird das Servlet direkt mit dem '@WebServlet'-Befehl deklariert. Dieses Servlet wird nun noch über das 'urlPattern' einem Link zu geteilt. Wenn der Link aufgerufen wird, wird das Servlet aufgerufen und die Methoden, welche hier definiert sind, werden ausgeführt. Es wurden 2 Servlet für jeweils ein Objekt angelegt, ein Servlet für das Erstellen und Aktualisieren und ein Servlet zum Löschen. Das Servlet zum Erstellen bzw. Aktualisieren der Objekte, enthalt 4 Methoden: 'doPost', 'doGet', 'sendRedirect', 'getWiring'. Die 'doPost'-Methode ruft nur die 'doGet'-Methode auf, welche mit einem 'try−catch'-Statement geschützt wird. In der 'doGet'-Methode wird nun eine neue Resource erstellt und durch die Methode 'getWiring' werden nun die Attribut des Objektes gesetzt und die Daten über den Request mit Hilfe der ID, welche auf der Webseite festgesetzt wurde, von der Webseite geholt. Nachdem nun das Objekt zurückgegeben wird, wird nun geprüft, ob die Resource bereits schon angelegt wurde, da entweder die Resource nur aktualisiert werden muss oder evtl. auch neu angelegt werden muss. Zum Schluss wird eine Antwort zurückgeschickt, welche in der Methode 'sendRedirect' erstellt wird. In der Methode wird mit der aktuellen ResourceID ein Link zusammengestellt. Das bewirkt, dass dann nun die bearbeitete Resource angezeigt wird. Das Servlet zum Löschen einer Resource ist ebenfalls gleich aufgebaut, jedoch wird auf die allgemeine Übersicht zurückgeleitet

## 4.4 Oberfläche

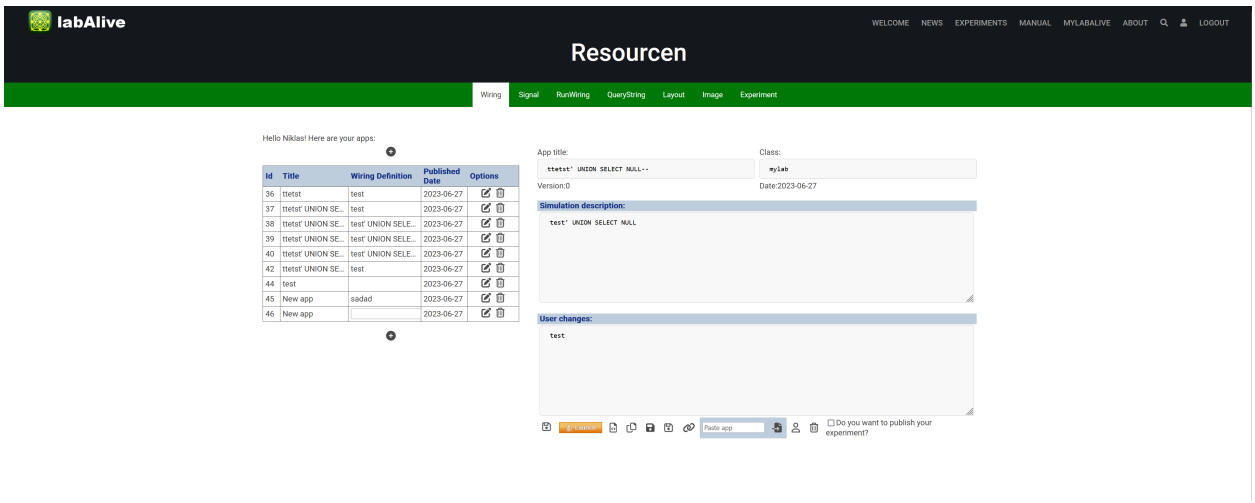


Abbildung 4.11: Wiring-Oberfläche

Zum Schluss wurde die Oberfläche erstellt, welche jedoch schon an einer anderen Stelle bereits erstellt wurde. Hier wurde die alte Jsp verwendet und mit den aktuellen Daten aktualisiert. Bei der einen oder anderen Jsp mussten noch ein paar Bedienelemente hinzugefügt werden.

# 5 Web Security

Dieses Kapitel beschäftigt sich mit der Websecurity der Webanwendung. Hierbei steht der automatische Scan, die SQL-Injection und das Cross-Site-Scripting im Mittelpunkt. Zuerst wird der automatische Scan erklärt und analysiert. Danach wird auf die SQL-Injection und zum Schluss auf das Thema Cross-Site-Scripting eingegangen.

## 5.1 ZAP-Scan

Als Erstes wurde mit Hilfe des Tools OWASP Zed Attack Proxy (ZAP) ein vorab Scan für die Webanwendung durchgeführt. Dieses Tool scannt die Webanwendung nach kritischen Sicherheitslücken. Das Tool scannt auch alle Seiten und listet die Seiten und die gefundenen Sicherheitslücken auf und zwar je nachdem wie schwerwiegend diese sind und markiert diese in unterschiedlichen Farben.

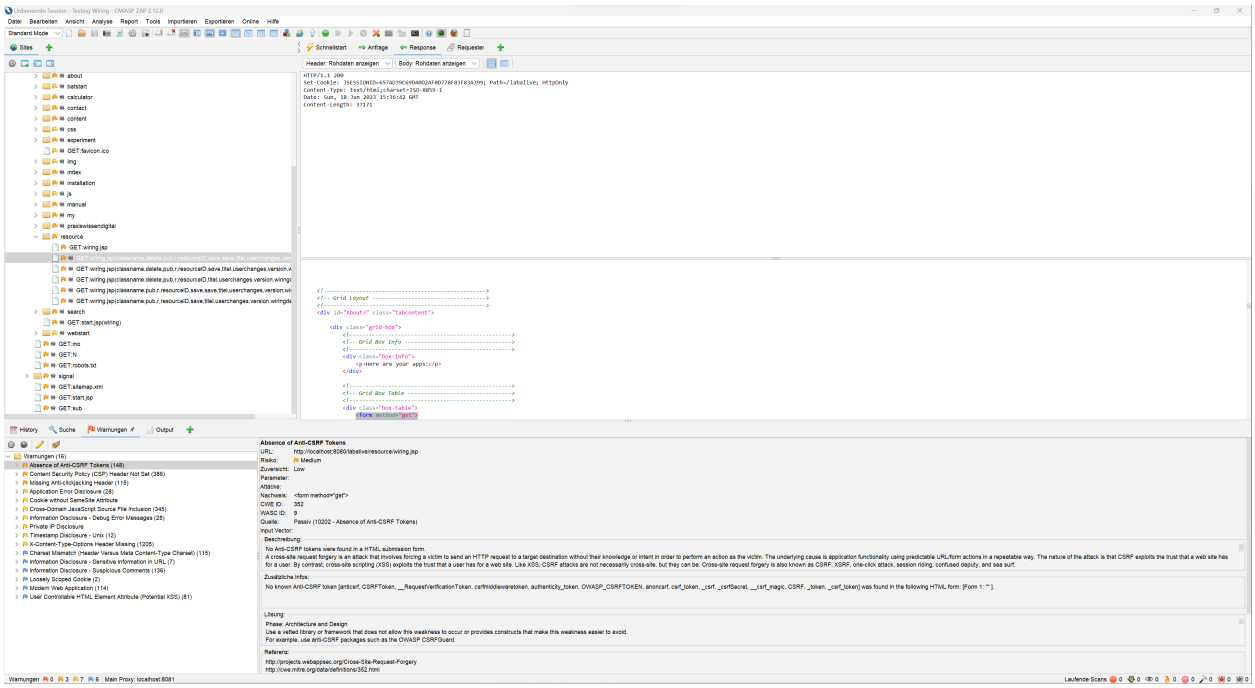

Abbildung 5.1: ZAP-Fenster

Ein Ergebnis des Scans der labeAlive-Webanwendung ist in der Abbildung 5.1 dargestellt. Es wurde kein hohes Risiko, sondern nur ein Medium-Risiko erkannt. Das Schwerwiegendste ist die Abwesenheit von Anti-CSRF-Tokens. CSRF steht für den Begriff ‚Cross-Site Request Forgery'. Hierbei handelt es sich um eine Art von Webangriff, bei dem ein Angreifer den Browser eines ahnungslosen Benutzers

dazu bringt, ungewollte Aktionen auf einer anderen Website auszuführen, auf der der Benutzer authentifiziert ist. Diese Angriffe können dazu führen, dass der Angreifer im Namen des Benutzers Aktionen ausführt, wie z.B. das Ändern von Passwörtern, das Senden von Nachrichten oder das Durchführen von Geldtransaktionen. Die Token verhindern nun diese Angriffe dadurch, dass das Token beim Absenden des Formulars oder der Aktion überprüft wird, um sicherzustellen, dass die Anfrage vom richtigen Benutzer stammt. Auch vermisst das Tool die Content Security Policy (CSP). Die CSP ist ein Mittel um Cross-Site-Scripting zu verhindern, welche im Kapitel Cross-Site-Scripting vorgestellt werden. Der letzte etwas kritische Punkt, der hier aufgelistet wird, ist der nicht vorhandene Anti-Clickjacking-Header. Das ist eine Sicherheitsmaßnahme, die verwendet wird, um Clickjacking-Angriffe zu verhindern. Bei einem Clickjacking-Angriff wird eine Webseite in einem unsichtbaren Rahmen (iFrame) platziert, um Benutzer dazu zu verleiten, ungewollte Aktionen auf der Webseite auszuführen, ohne es zu bemerken. Der Header kann in den HTTP-Antworten einer Website gesetzt werden und enthält eine Richtlinie, die dem Browser mitteilt, wie mit dem Laden der Seite in einem iFrame umgegangen werden soll. Somit verhindert diese Gegenmaßnahme Clickjacking-Angriffe. Jedoch ist der Scan nicht genug und gibt nur einen groben Überblick, welche Sicherheitslücken existieren könnten. Um weitere Sicherheitslücken von dieser Webanwendungen zu analysieren, wird in dem folgenden Kapitel dieser Arbeit auf SQL-Injections und Cross-Site-Scripting eingegangen.

## 5.2 SQL-Injection

Im Folgenden wird näher auf die SQL-Injection eingegangen und die Frage, was eine SQL-Injection überhaupt ist, beantwortet. SQL-Injection ist eine Sicherheitslücke in Webanwendungen die Datenbanken verwenden. Es handelt sich um eine Angriffstechnik, bei der ein Angreifer böswillige SQL-Befehle in eine Eingabeaufforderung oder ein Webformular eingibt, mit dem Ziel, die Datenbankabfragen der Anwendung zu manipulieren und unerlaubten Zugriff auf die Datenbank zu erlangen. Bei unsachgemäßer Behandlung der Benutzereingaben können Angreifer speziell gestaltete Zeichenfolgen einschleusen, die von der Anwendung nicht ordnungsgemäß überprüft oder gefiltert werden. Diese manipulierten Zeichenfolgen werden dann als Teil eines SQL-Befehls interpretiert und an die Datenbank gesendet. Die potenziellen Auswirkungen von SQL-Injection können schwerwiegend sein. Ein Angreifer kann dadurch in die Lage versetzt werden unerlaubt Datenbankinhalte zu lesen, zu ändern, zu löschen, Benutzerrechte zu eskalieren, die Ausführung von Befehlen auf dem Datenbankserver auszulösen oder sogar die gesamte Anwendung zu unterlaufen. Es gib dabei mehrere Möglichkeiten an die Datenbankinhalte oder an die Informationen über die Datenbank zu kommen. In dieser Arbeit werden 4 Beispiele an SQL-Injections betrachtet:

- Versteckte Daten abrufen,
- UNION-Angriffe,
- Untersuchen der Datenbank,
- Blinde SQL-Injection.

[10]

### 5.2.1 Versteckte Daten abrufen

Um nun an versteckte Daten zu kommen, wie zum Beispiel andere Nutzerdaten aus der Nutzertabelle, wird eine SQL-Injection mit Hilfe des Login-Fenster ausgeführt. Dazu wurde das Programm Burp Suite mit seinem Proxy und dem Repeater genutzt. Mit Hilfe des Proxy werden die Verbindungen unterbrochen, und jeder einzelne Request kann dabei ausgewertet werden. Nachdem nun das Login-Fenster umgesetzt wurde, wurde mit der ersten einfachen SQL-Injection sowohl das 'Login' als auch das 'Register' getestet. Dazu wird der Request, welcher geschickt wird um sich einzuloggen, genommen und an den Repeater weitergegeben. Mit Hilfe des Repeaters kann der Request abgeändert werden um immer wieder neue Angriffe bzw. Requests loszuschicken. Gleichzeitig wird die Response des Requests angezeigt und ausgewertet.

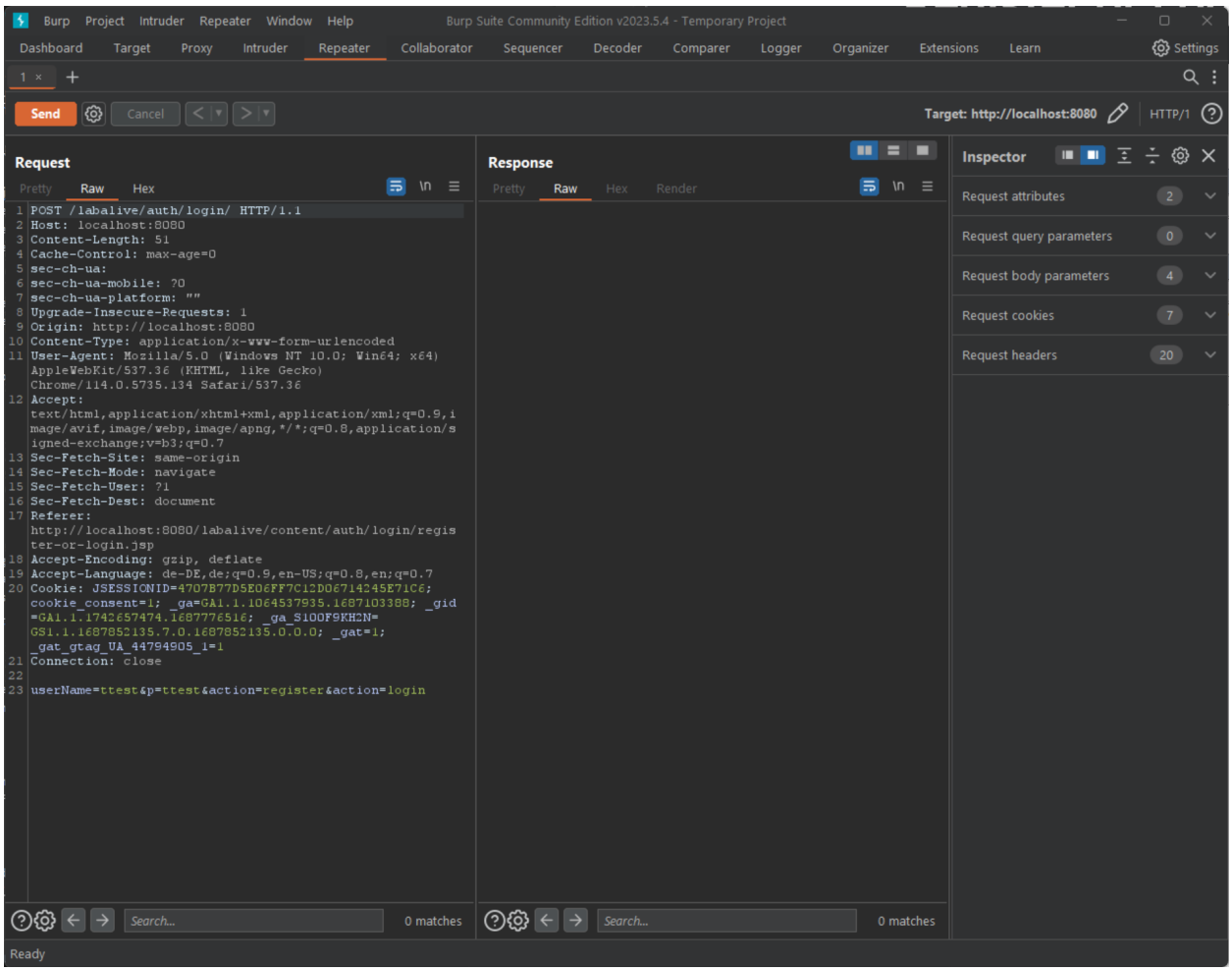


Abbildung 5.2: Burp-Suie-Intruder

Nun können die ersten Angriffe ausgeführt werden. Dazu wird zum Beispiel beim Username hinter dem 'test' ein "+ OR+1=1–-' eingefügt und dann der Request abgeschickt. Wenn nun keine Schutzmechanismen eingebaut sind, bewirkt das die folgende Abfrage:

```
SELECT * FROM T_MyUser WHERE Username = 'test' OR 1=1--'
```

SQL

Hierbei ist wichtig, dass alles nach der Doppelstrichfolge von SQL ignoriert wird, da diese den Kommentarindikator in SQL darstellen. Das bedeutet nun, dass alle Benutzer mit dem Namen 'test' ausgegeben werden. Um alle Daten zubekommen fügt ein Angreifer die Erweiterung 'OR 1=1--' an. Dieser Ausdruck bewirkt, dass entweder alle Einträge mit dem Benutzernamen 'test' oder alle Einträge, bei denen 1=1 ist, ausgegeben werden. Dadurch, dass 1=1 immer wahr ist, werden bei ungeschützten Webanwendungen alle Benutzernamen zurückgegeben. [11]
Dadurch, dass aber bereits bei dieser Webanwendung sowohl beim Login/Register als auch beim Anlegen von Resourcen Vorkehrungen getroffen wurden, konnten mit dieser Methode keine zusätzlichen Informationen aus der in dieser Arbeit erstellten Datenbank geholt werden. Vielleicht funktioniert die nächste Art von SQL-Injection.

### 5.2.2 UNION-Angriffe

Da bis jetzt kein Angriff zum Erfolg geführt hat, wurden die nächste Art von SQL-Injection ausprobiert, und zwar mit dem Schlüsselwort UNION. Mit dem Schlüsselwort können Daten aus anderen Tabellen abgefragt werden. Zum Beispiel wird eine WiringResource aus der Datenbank abgefragt. Innerhalb dieser Abfrage kann mit Hilfe von UNION eine zweite Abfrage gestartet werden, welche zum Beispiel in einer anderen Tabelle nach Username und Passwort sucht. Damit aber diese Abfrage funktioniert müssen folgende Anforderungen erfüllt sein: Die einzelnen Abfragen müssen die gleiche Anzahl an Spalten zurückgeben und die Anzahl der Datentypen in jeder Spalte müssen zwischen den einzelnen Abfragen kompatibel sein. Bevor nun ein UNION-Angriff durchgeführt werden kann, muss zuerst herausgefunden werden wie viele Spalten die ursprüngliche Abfrage hat. Des Weiteren muss auch bekannt sein, welche Spalten den richtigen Datentyp zum Speichern der UNION-Abfragen enthalten. Als Erstes wird versucht mit Hilfe einer 'ORDER BY 1--' Abfrage oder einer 'UNION SELECT NULL--' herauszufinden, wie viel Spalten die Abfrage zurückgibt. Dabei wird im ersten Fall einfach die Zahl erhöht oder im zweiten Fall wird einfach Komma-getrennt eine weitere 'NULL' hinzugefügt. Dies wird so lange durchgeführt bis kein Error mehr angezeigt wird. Sobald dieser Punkt erreicht ist, ist dem Angreifer bekannt wie viele Spalten bei dieser Abfrage zurückgegeben werden. Der nächste Schritt für den Angreifer ist es nun herauszufinden, welche Spalten den für ihn nützlichen Datentyp enthalten, damit seine Ergebnisse gespeichert werden können. Das passiert wieder mit dem diesem Ausdruck: 'UNION SELECT 'a',NULL,NULL--'. Mit dem 'a' wird nun getestet, ob die erste Spalte, die zurückgegeben wird, den Datentyp 'String' enthält. Wenn nun kein Fehler auftritt und die Antwort der Anwendung den eingefügten Zeichenfolgenwert enthält, weiß der Angreifer in welcher Spalte er seine Daten speichern kann. Zum Schluss führt der Angreifer einen 'UNION'-Angriff durch und kann sich zum Beispiel alle Benutzer und deren Passwörter ausgeben lassen. [12]
Aufgrund der bereits erwähnten Vorkehrungen, könnte ein 'UNION'-Angriff bei der erstellten Applikation nicht erfolgreich durchgeführt werden. Dies hat auch etwas mit dem Aufbau der Webanwendung zu tun. Da nur beim Server-Start eine 'SELECT'-Abfrage ausgeführt wird, ist es schwierig weitere SQL-

Injectionen einzuschleusen. Auch beim Login funktionierte der Angriff nicht. Warum genau dies hier nicht funktionierte, wird im Unterkapitel Vorkehrungen erklärt.

### 5.2.3 Untersuchen der Datenbank

Um doch noch erfolgreiche Angriffe auszuführen, kann ein Angreifer noch Informationen über die Datenbank an sich holen, wie zum Beispiel Sprache und Version. Dazu wird ebenfalls wieder die UNION Abfrage hergenommen. Zum Beispiel wird die Version bei MySQL-Datenbank mithilfe von 'SELECT @@version' abgefragt. Wenn dieser Angriff erfolgreich ist, ist dem Angreifer bekannt welche Datenbank im Hintergrund läuft. Nun kann der Angreifer mehr Information über die Inhalte der Datenbank herausfinden. Dazu nutzt er folgenden Befehl: 'SELECT * FROM information_schema.tables' Dabei wird ausgegeben wie viele Tabellen die Datenbank enthält und wie diese Tabellen heißen. Danach kann der Angreifer die einzelnen Spalten der Tabelle abfragen.[13]

```
2 SELECT * FROM information_schema.columns WHERE table_name = 'T_MyUser'
```

SQL

Auch diese Angriffe wurden auf die labeAlive-Webanwendung ausgeführt, jedoch waren diese auch wieder sehr erfolglos und es konnten hier keine Informationen über die Datenbank abgerufen werden.

### 5.2.4 Blind SQL-Injection

Die letzte Möglichkeit eine erfolgreiche SQL-Injection durchzuführen ist durch Auslösen bedingter Anweisungen Antworten der Datenbank zu erreichen. Hier wird mit folgenden Befehlen versucht eine wahre oder falsche Antwort zu erhalten, um zum Beispiel einen Nutzernamen oder ein Passwort herauszubekommen. Im Folgenden ein Beispiel dazu:

```
3 Cookie: user=a84847de-f728-4039-9571-855286071698' AND (SELECT 'a' FROM users ↩
    WHERE username='Niklas')='a
```

SQL

Nachdem hier ein Cookie gesetzt ist, welcher auch in der Datenbank gespeichert ist, kann mit bestimmten Anweisungen und Fragen an die Datenbank, Informationen der Webanwendung entlockt werden. In obigen Beispiel wird nachgefragt, ob ein Nutzer namens Niklas existiert. Falls nun dieser existiert, kann der Angreifer mit weiteren Fragen das Passwort des Nutzers oder des Admins herausfinden. Beispiel:

```
4 Cookie: user=a84847de-f728-4039-9571-855286071698' AND (SELECT 'a' FROM T_MyUser↩
     WHERE Username='Niklas' AND LENGTH(password)>3)='a
```

SQL

[14]

### 5.2.5 Vorkehrungen

Um SQL-Injection zu verhindern, kann folgende bewährte Praktik im Code angewendet werden: parametrisierte Abfragen. Das bedeutet, dass statt Benutzereingaben direkt in die SQL-Abfragen einzufügen, parametrisierte Abfragen verwendet werden.

```
1  String preparedStat = "INSERT INTO R_Resource (Titel, SaveDate, Description, ↩
       UserID, OriginResourceID, PublishedStatus, ResourceType, UUID ) VALUES ( "
2    + "?  , "//Titel
3    + "?  , "//Description
4    + "?  , "//OriginResourceID
5    + "?  , "//PublishedStatus
6    + "?  , "//UserID
7    + "?  , "//ResourceType
8    + "?  , "//SaveDate
9    + "?  )";//UUID
10 Object[] parameterList = {titel, saveDate, description, userID, originResourceID↩
       , publishedStatus, resourceType, uuid};
11 DBUpdateCommand cmd = new DBUpdateCommand(preparedStat, parameterList);
```

Java

Dadurch werden die Benutzereingaben als separate Parameter behandelt und nicht als Teil der eigentlichen SQL-Abfrage interpretiert. Dies bietet einen besseren Schutz vor SQL-Injection-Angriffen. [15]

## 5.3 Cross-Site-Scripting

Cross-Site-Scripting (XSS) ist eine Sicherheitslücke, die in Webanwendungen auftritt und es einem Angreifer ermöglicht, schädlichen Code (in der Regel JavaScript) in die angezeigten Webseiten einzufügen, die von anderen Benutzern besucht werden. Diese Art von Angriffen können auftreten, wenn eine Webanwendung unzureichend validiert oder filtert, welche Inhalte von Benutzern bereitgestellt und in die Webseiteninhalte eingefügt werden. Da nun die Webanwendung mehrere Eingabemöglichkeiten besitzt und auch andere Nutzer die Resource sehen können, ist es wichtig, dass diese Anwendung auch gegen Cross-Site-Scripting geschützt wird. Es gibt dabei 2 Arten von Cross-Site-Scripting. Diese werden in den nächsten Unterkapiteln näher vorgestellt. [16]

### 5.3.1 Reflected XSS

Hierbei injiziert der Angreifer schädlichen Code in eine URL oder ein Webformular. Wenn ein anderer Benutzer diesen Link aufruft oder das Formular ausfüllt, wird der schädliche Code vom Webserver zurückgegeben und im Browser des Benutzers ausgeführt. Ein Angreifer kann nun ganz einfach Testen, ob Cross-Site-Scripting möglich ist in dem er '<script>alert("test")</script>' in die Eingabemöglichkeiten eingibt. Wenn er nun den Request losschickt und die Anwendung den Text wieder in der Antwort enthält, wird dieser Code ausgeführt und der Browser zeigt eine Meldung mit den Worten "test". Damit ist

nun bewiesen, dass auf der Seite Cross-Site-Scripting möglich ist. Nun kann der Angreifer mit Hilfe von Java-Code einiges anstellen. Der Angreifer kann nun Cookies stehlen, Passwörter speichern oder Cross-Site-Request-Forgery ausführen. [17]
Nachdem dies auf der Webseite getestet wurde, konnte erfolgreich einen 'Alert' auf der Webseite ausgeführt werden. Auch das Stehlen der Cookies funktioniert ohne Probleme. Das Gefährliche hierbei ist, dass der Admin alle User sieht und somit sein Cookie geklaut wird. Das ermöglicht dem Angreifer auch viel Schadcode in die Resourcen eines Benutzers zu verstecken. Dabei kann dieser auch Passwörter der Nutzer für weitere Webanwendungen erspähen.

### 5.3.2 Stored XSS

Bei dieser Art von Angriff wird der schädliche Code auf dem Webserver gespeichert und auf allen Seiten angezeigt, die den entsprechenden Inhalt anzeigen. Dies kann zum Beispiel in Kommentarfunktionen oder Foren auftreten, in denen Benutzer Inhalte veröffentlichen können. Hier ist der Fall ähnlich gelagert wie bei dem Reflected XSS. Jedoch wird hier der Script-Code in der Webanwendung gespeichert. Dadurch, dass die Webanwendung die Seite bzw. die Eingabe automatisch speichert, wird auch der Script-Code gespeichert. Wenn nun ein anderer Benutzer diese Seite besucht, wird der Script-code erneut ausgeführt und die Daten des Benutzers an den Angreifer weitergeleitet. [18]
Da auch die vorliegende Webanwendung die Eingaben speichert, ist hier auch eine stored XSS"möglich. Auch die Gefahren hierbei sind genau die Gleichen wie bei der "Reflected XSS".

### 5.3.3 Vorkehrungen

Um nun Cross-Site-Scripting zu verhindern, gibt es nun mehrere Möglichkeiten:

- Eingabevalidierung,
- Ausgabevalidierung,
- Content-Security-Policy,
- HTTP-Only-Cookies.

Bei der Eingabevalidierung werden alle Benutzereingaben überprüft und gefiltert. Damit wird sichergestellt, dass diese keine schädlichen Zeichen oder Skripttags enthalten. Mit der Ausgabevalidierung werden alle Daten überprüft, welche Daten auf der Webseite angezeigt werden und das einschließlich der Benutzereingaben, Datenbankabfragen und anderen dynamischen Inhalten. Ein weiterer Schutz ist die Content-Security-Policy (CSP). Damit kann festlegt werden von welchen Quellen bestimmte Resourcen wie Skripte, Stylesheets oder Bilder auf der Webseite geladen werden dürfen. Dies hilft dabei, nicht vertrauenswürdigen Code an der Ausführung zu hindern. Ein weiterer Schutz ist das ‚HttpOnlyFlag' für Cookies. Das verhindert zum Beispiel, dass JavaScript auf diese Cookies zugreifen kann. Somit kann auch das Risiko von XSS-Angriffen verringert werden, da gestohlene Cookies weniger missbraucht werden können. [19]

# 6 Fazit

Die Weiterentwicklung der labAlive-Webanwendung mit einem Fokus auf Websecurity ist von zentraler Bedeutung, um den steigenden Bedrohungen im digitalen Raum entgegenzuwirken und ein sicheres und vertrauenswürdiges Nutzungserlebnis zu gewährleisten. Durch die Implementierung robuster Sicherheitsmaßnahmen kann labAlive die Vertraulichkeit sensibler Informationen bewahren, die Integrität der Anwendung gewährleisten und die Verfügbarkeit für die Benutzer sicherstellen. Darüber hinaus stärkt die kontinuierliche Weiterentwicklung von labAlive die Widerstandsfähigkeit gegenüber neuen Angriffstechniken und hält die Anwendung auf dem neuesten Stand der Sicherheitsstandards. Die Benutzer können sich darauf verlassen, dass ihre Daten geschützt sind und ihre Privatsphäre gewahrt bleibt, während sie die Vorteile der vielseitigen Funktionen und Möglichkeiten von labAlive nutzen. Die Investition in die Weiterentwicklung der Websecurity von labAlive ist daher unerlässlich, um den wachsenden Anforderungen gerecht zu werden und den Benutzern eine sichere und zuverlässige Plattform für ihre Aktivitäten bereitzustellen.

## 6.1 Ausblick

Eine Weiterentwicklung dieser Webanwendung kann in Zukunft in vielfältiger Weise stattfinden. Die Webanwendung kann zum Beispiel damit erweitert werden, dass Benutzer sich gegenseitig ihre Resourcen teilen oder veröffentlichen. Dabei kann ein Benutzer sein erschaffenes Werk einem Administrator zur Veröffentlichung vorschlagen. Der Administrator kann nun die Resource überprüfen und diese dann veröffentlichen. Des Weiteren können bezüglich Websecurity noch weitere und tiefergehende Tests ausgeführt werden. Hier könnte in Zukunft gegen Cross-site request forgery, Clickjacking, Access Controll und File Vulnerabilities Tests durchgeführt werden. Gerade gegen die Sicherheitslücken bei Dateien bzw. bei einem Datei-Upload können noch mehrere Sicherheitsmechanismen implementiert werden.

# Anhang

# Abbildungsverzeichnis

# Quellcodeverzeichnis



```
1  package resource.data;
2
3  import de.labAlive.web.util.Utils;
4
5  public class WiringResource extends Resource {
6
7      private long wiringID = -1;
8      private String classname;
9      private String wiringDefinition;
10     private String userChanges;
11     private String title;
12     private int version;
13
14     public static WiringResource create(long wiringID) {
15         WiringResource wiring = new WiringResource();
16         wiring.setWiringID(wiringID);
17         return wiring;
18     }
19
20     public WiringResource() {
21         super();
22     }
23
24     public long getWiringID() {
25         return wiringID;
26     }
27
28     public void setWiringID(long wiringID) {
29         this.wiringID = wiringID;
```

```
30      }
31
32      public String getClassname() {
33          return classname;
34      }
35
36      public void setClassname(String classname) {
37          this.classname = classname;
38      }
39
40      public String getWiringDefinition() {
41          return wiringDefinition;
42      }
43
44      public void setWiringDefinition(String wiringDefinition) {
45          this.wiringDefinition = wiringDefinition;
46      }
47
48      public String getUserChanges() {
49          return userChanges;
50      }
51
52      public void setUserChanges(String userCahnges) {
53          this.userChanges = userCahnges;
54      }
55
56      public String getTitle() {
57          return title;
58      }
59
60      public void setTitle(String title) {
61          this.title = title;
62      }
63
64      public int getVersion() {
65          return version;
66      }
67
68      public void setVersion(int version) {
69          this.version = version;
70      }
71
72      @Override
73      public String getResourceDetailsJsp() {
74          // TODO create wiringdetails.jsp for public view
75          return "wiringdetails.jsp";
```

```
76      }
77
78    public String getFilename4JnlpBat() {
79      // replace forbidden characters in filename * . " / \ [ ] : ; | , < ↩
            > etc.
80      return title.replace(" ", "-");
81    }
82
83    // contains no user!
84    public String getParameters() {
85      return "id=" + getResourceID() +
86          getParameter("c", getClassname()) +
87          getParameter("v", "" + version) +
88          getParameter("t", title) +
89          getParameter("w", getWiringDefinition()) +
90          getParameter("p", getUserChanges()) +
91          getParameter("pub", "" + getPublishedStatus());
92    }
93
94    private String getParameter(String key, String unencodedValue) {
95      if (unencodedValue == null || "".equals(unencodedValue))
96        return "";
97      return "&" + key + "=" + Utils.urlEncode(unencodedValue);
98    }
99
100 }
```

Quellcode 6.1: WiringResource

```
1  package resource.db.impl;
2
3  import java.sql.ResultSet;
4  import java.sql.SQLException;
5  import java.util.ArrayList;
6  import java.util.Collection;
7  import java.util.UUID;
8
9  import myLabAlive.common.exceptions.DBException;
10 import myLabAlive.dbutils.DBSelectCommand;
11 import myLabAlive.dbutils.DBUpdateCommand;
12 import resource.data.ExperimentResource;
13 import resource.data.ImageResource;
14 import resource.data.LayoutResource;
15 import resource.data.QueryStringResource;
16 import resource.data.Resource;
17 import resource.data.RunWiringResource;
```

```
18 import resource.data.SignalResource;
19 import resource.data.WiringResource;
20
21 //Kommentar
22
23 public class ResourceDBImpl {
24
25   @SuppressWarnings("unchecked")
26   public static Collection<Resource> getAllResources() throws ↩
       DBException {
27     DBSelectCommand cmd = getAllResourcesCmd();
28     return (Collection<Resource>) cmd.run();
29   }
30
31   private static DBSelectCommand getAllResourcesCmd() throws DBException↩
        {
32     String preparedStat = "SELECT * FROM R_Resource";
33     DBSelectCommand cmd = new DBSelectCommand(preparedStat) {
34       protected Object handleResultSet(ResultSet rs) throws Exception {
35         Collection<Resource> resources = new ArrayList<>();
36         while (rs.next()) {
37           Resource resource = createReturnResourceType(rs.getString("↩
               ResourceType"));
38           resource.setResourceID(rs.getLong("ResourceID"));
39           resource.setTitel(rs.getString("Titel"));
40           resource.setDescription(rs.getString("Description"));
41           resource.setOriginResourceID(rs.getLong("OriginResourceID"));
42           resource.setPublishedStatus(rs.getInt("PublishedStatus"));
43           resource.setUserID(rs.getLong("UserID"));
44           resource.setResourceType(rs.getString("ResourceType"));
45           resource.setSaveDate(rs.getString("SaveDate"));
46           resources.add(resource);
47         }
48
49         return resources;
50       }
51
52
53
54       private Resource createReturnResourceType(String type) {
55         switch (type) {
56           case "signal": {
57             return new SignalResource();
58           }
59           case "image": {
60             return new ImageResource();
```

```
61            }
62            case "experiment": {
63              return new ExperimentResource();
64            }
65            case "wiring": {
66              return new WiringResource();
67            }
68            case "runWiring": {
69              return new RunWiringResource();
70            }
71            case "queryString": {
72              return new QueryStringResource();
73            }
74            case "layout": {
75              return new LayoutResource();
76            }
77            default:
78              throw new IllegalArgumentException("Unexpected value: " + ↩
                    type);
79          }
80        }
81      };
82      return cmd;
83    }
84
85
86    public static Resource createResource(Resource resource) throws ↩
          DBException {
87      String description = resource.getDescription();
88      String titel = resource.getTitel();
89      long originResourceID = resource.getOriginResourceID();
90      int publishedStatus = resource.getPublishedStatus();
91      long userID = resource.getUserID();
92      String resourceType = resource.getResourceType();
93      String saveDate = resource.getSaveDate();
94      String uuid = UUID.randomUUID().toString();
95
96          String preparedStat = "INSERT INTO R_Resource (Titel, SaveDate, ↩
                Description, UserID, OriginResourceID, PublishedStatus, ↩
                ResourceType, UUID ) VALUES ( "
97                    + "? , "
98          + "? , "
99                    + "? , "
100                   + "? , "
101         + "? , "
102         + "? , "
```

```
103         + "? , "
104             + "? )";
105         Object[] parameterList = {titel, saveDate, description, userID, ↩
                originResourceID, publishedStatus, resourceType, uuid};
106       DBUpdateCommand cmd = new DBUpdateCommand(preparedStat, ↩
              parameterList);
107       System.out.println(preparedStat);
108       cmd.run();
109
110       long resourceID = 1;
111       try{
112         resourceID = getResourceIdByUUID(uuid);
113       } catch (Exception e){
114         e.printStackTrace();
115       }
116           resource.setResourceID(resourceID);
117       return resource;
118     }
119
120
121       public static void updateResource(Resource resource) throws ↩
              DBException {
122           String preparedStat = "UPDATE R_Resource SET Titel = ?, SaveDate↩
                  = ?, Description = ?, PublishedStatus = ?, ResourceType = ?↩
                  WHERE ResourceID = ?";
123           Object[] parameterList = {
124           resource.getTitel(),
125           resource.getSaveDate(),
126           resource.getDescription(),
127           resource.getPublishedStatus(),
128           resource.getResourceType(),
129           resource.getResourceID()};
130           DBUpdateCommand cmd = new DBUpdateCommand(preparedStat, ↩
                  parameterList);
131           cmd.run();
132       }
133
134     public static Long getResourceIdByUUID(String uuid) throws DBException↩
          {
135       String preparedStat = "SELECT * FROM R_Resource WHERE UUID = ?";
136       Object[] parameterList = {uuid};
137       DBSelectCommand cmd = new DBSelectCommand(preparedStat, ↩
              parameterList) {
138         protected Object handleResultSet(ResultSet rs) throws SQLException↩
              {
139           rs.next();
```

```
140         // String desc = rs.getString("Description");
141         return rs.getLong("ResourceID");
142       }
143     };
144     return (Long) cmd.run();
145   }
146 
147   public static int deleteResource(Resource resource) throws DBException↩
        {
148     String selectQuery = "DELETE FROM R_Resource WHERE ResourceID = ?";
149     Object[] parameterList = {resource.getResourceID()};
150     System.out.println(selectQuery);
151     DBUpdateCommand cmd = new DBUpdateCommand(selectQuery, parameterList↩
          );
152     return (Integer) cmd.run();
153   }
154 }
```

Quellcode 6.2: Resource Datenbank-adapter

```
1  package resource.db;
2  
3  import resource.data.ResourceContainer;
4  import resource.data.WiringResource;
5  import resource.provider.ResourceProvider;
6  import resource.provider.ResourceProviderInitializer;
7  
8  import javax.servlet.http.HttpServletRequest;
9  import java.util.Collection;
10 
11 public class WiringSelector4Jsp {
12     static int init = 0;
13 
14     public static Collection<ResourceContainer> getAllWirings(↩
           HttpServletRequest request) {
15         if(init == 0){
16             ResourceProviderInitializer.init();
17             init = 1;
18         }
19         return ResourceProvider.getAllWirings(request);
20     }
21 }
```

Quellcode 6.3: Wiring Selector for Jsp

```
1  package resource.db;
2  
```

```
3  import myLabAlive.common.exceptions.DBException;
4  import resource.data.ResourceContainer;
5  import resource.data.WiringResource;
6  import resource.db.impl.R_wiringDBImpl;
7  import resource.provider.ResourceProvider;
8
9  import javax.servlet.http.HttpServletRequest;
10 import java.util.Collection;
11
12 public class WiringDB4Servlet {
13
14
15     public static Collection<ResourceContainer> getAllWirings(↩
           HttpServletRequest request) {
16         return ResourceProvider.getAllWirings(request);
17     }
18
19
20     public static WiringResource getWiringByResourceId(long resourceId) ↩
           throws DBException {
21         return (WiringResource) ResourceProvider.getResourceById(↩
               resourceId).getResource();
22     }
23
24
25     public static WiringResource createWiring(WiringResource wiring) ↩
           throws DBException {
26         wiring = (WiringResource) ResourceDBInternal.createResource(↩
               wiring);
27         wiring = R_wiringDBImpl.createWiring(wiring);
28         return wiring;
29     }
30
31
32
33     public static void updateWiring(WiringResource wiring) throws ↩
           DBException {
34         R_wiringDBImpl.updateWiring(wiring);
35         ResourceProvider.updateResource(wiring);
36     }
37
38     public static void deleteWiring(WiringResource wiring) throws ↩
           DBException {
39         R_wiringDBImpl.deleteWiring(wiring);
40         ResourceDBInternal.deleteResource(wiring);
41     }
```

```
42
43 }
```

Quellcode 6.4: Wiring DB for Servlet

```
1  package resource.servlet;
2
3  import resource.auth.SessionAccess;
4  import de.labAlive.web.util.TextUtils;
5  import myLabAlive.common.User;
6  import myLabAlive.common.exceptions.DBException;
7  import resource.auth.AccessCheck;
8  import resource.data.WiringResource;
9  import resource.db.WiringDB4Servlet;
10
11 import javax.servlet.ServletException;
12 import javax.servlet.annotation.WebServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15 import java.io.IOException;
16 import java.time.LocalDate;
17
18 @WebServlet(urlPatterns = { "/wiring/createupdate/" })
19 public class WiringServlet extends javax.servlet.http.HttpServlet ↩
       implements javax.servlet.Servlet{
20
21     private static final long serialVersionUID = -4694875471965182952L;
22
23     protected void doPost(HttpServletRequest request, ↩
           HttpServletResponse response){
24         try {
25             doGet(request, response);
26         } catch (ServletException e) {
27             throw new RuntimeException(e);
28         } catch (IOException e) {
29             throw new RuntimeException(e);
30         }
31     }
32
33
34     protected void doGet(HttpServletRequest request, HttpServletResponse↩
           response)throws ServletException, IOException{
35         User user = SessionAccess.getUser(request);
36         if (!AccessCheck.checkAccessForUser(request, user)){
37             response.sendRedirect("/labalive/auth/login/");
38             return;
```

```
39          }
40          WiringResource wiringFromRequest = getWiring(request);
41          wiringFromRequest.setSaveDate(LocalDate.now().toString());
42          wiringFromRequest.setResourceType("wiring");
43          if(wiringFromRequest.getResourceID() >=0){
44              try{
45                  WiringDB4Servlet.updateWiring(wiringFromRequest);
46              } catch (DBException e1){
47                  e1.printStackTrace();
48              }
49              sendRedirect(request, response, wiringFromRequest);
50          } else {
51              try {
52                  wiringFromRequest.setUserID(user.getUserId());
53                  wiringFromRequest.setVersion(0);
54                  WiringResource wiringResource = WiringDB4Servlet.↩
                        createWiring(wiringFromRequest);
55                  sendRedirect(request, response, wiringResource);
56              } catch (DBException e){
57                  e.printStackTrace();
58              }
59          }
60      }
61
62
63      private void sendRedirect(HttpServletRequest request, ↩
            HttpServletResponse response, WiringResource wiring) {
64          try {
65              response.sendRedirect(request.getContextPath() + "/resource/↩
                    wiring.jsp?resourceID=" + wiring.getResourceID());
66          } catch (IOException e) {
67              e.printStackTrace();
68          }
69      }
70
71      public static WiringResource getWiring(HttpServletRequest request) {
72          WiringResource wiring = new WiringResource();
73
74          String resourceID_str = request.getParameter("resourceID");
75          String classname = request.getParameter("classname");
76          String version = request.getParameter("version");
77          String title = request.getParameter("titel");
78          String oldwiring = request.getParameter("s");
79          if (oldwiring != null) {
80              System.err.println("request.getParameter(\"s\") noch gesetzt↩
                    !");
```

```
81          }
82          String wiringdef = request.getParameter("wiringdef");
83          String userchanges = request.getParameter("userchanges");
84
85          if(!TextUtils.isEmpty(resourceID_str)){
86              long resourceID = Long.parseLong(resourceID_str);
87              wiring.setResourceID(resourceID);
88          }
89
90          wiring.setClassname(classname);
91          wiring.setWiringDefinition(wiringdef);
92          wiring.setUserChanges(userchanges);
93          wiring.setTitle(title);
94          wiring.setVersion(Integer.parseInt(version));
95
96          return wiring;
97      }
98  }
```

Quellcode 6.5: Wiring Servlet

```
1  package resource.servlet;
2
3
4  import javax.servlet.ServletException;
5  import javax.servlet.annotation.WebServlet;
6  import javax.servlet.http.HttpServletRequest;
7  import javax.servlet.http.HttpServletResponse;
8
9  import myLabAlive.common.User;
10 import resource.auth.AccessCheck;
11 import resource.auth.SessionAccess;
12 import resource.data.WiringResource;
13 import resource.db.WiringDB4Servlet;
14 import resource.db.impl.ResourceDBImpl;
15
16 import java.io.IOException;
17
18 @WebServlet(urlPatterns = { "/wiring/delete/" })
19 public class WiringServlet4Delete extends javax.servlet.http.HttpServlet↩
       implements javax.servlet.Servlet{
20
21     private static final long serialVersionUID = -4694875471965182952L;
22
23     protected void doPost(HttpServletRequest request, ↩
           HttpServletResponse response)throws ServletException, ↩
```

```
        IOException{
24          doGet(request, response);
25      }
26
27      protected void doGet(HttpServletRequest request, HttpServletResponse↩
            response)throws ServletException, IOException{
28          User user = SessionAccess.getUser(request);
29          if (!AccessCheck.checkAccessForUser(request, user)){
30              response.sendRedirect("/labalive/auth/login/");
31              return;
32          }
33          WiringResource wiringFromRequest = getWiring(request);
34          try{
35              wiringFromRequest.setResourceType("wiring");
36              WiringDB4Servlet.deleteWiring(wiringFromRequest);
37              sendRedirect(request, response);
38          } catch (Exception e){
39              e.printStackTrace();
40          }
41      }
42
43      private void sendRedirect(HttpServletRequest request, ↩
            HttpServletResponse response) {
44          try {
45              response.sendRedirect(request.getContextPath() + "/resource/↩
                    wiring.jsp");
46          } catch (IOException e) {
47              e.printStackTrace();
48          }
49      }
50
51
52      public static WiringResource getWiring(HttpServletRequest request) {
53          WiringResource wiring = new WiringResource();
54
55          String resourceID = request.getParameter("resourceID");
56          wiring.setResourceID(Long.parseLong(resourceID));
57
58          return wiring;
59      }
60  }
```

Quellcode 6.6: Wiring Servlet for Delete

```
1   <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2            pageEncoding="ISO-8859-1"%>
```

```
3  <%@ page import="layout.*" %>
4  <%@ page import="layout.dblayer.*" %>
5  <%@ page import="ressource.details.*" %>
6  <%@ page import="ressource.*" %>
7  <%@ page import="java.util.*" %>
8  <%@ page import="de.labAlive.web.servlet.my.apps.URLgenerateServlet" %>
9  <%@ page import="myLabAlive.common.MyWiring" %>
10 <%@ page import="resource.data.WiringResource" %>
11 <%@ page import="myLabAlive.common.User" %>
12 <%@ page import="resource.auth.SessionAccess" %>
13 <%@ page import="de.labAlive.web.util.WiringUtils" %>
14 <%@ page import="de.labAlive.web.launch.LaunchMode" %>
15 <%@ page import="resource.db.WiringSelector4Jsp" %>
16 <%@ page import="resource.data.ResourceContainer" %>
17 <%@ page import="resource.provider.ResourceProviderInitializer" %>
18 <%@ page import="resource.db.SignalSelector4Jsp" %>
19
20
21
22 <%
23     User user = SessionAccess.getUser(request);
24     String salutation = null;
25     if(user == null){
26         response.sendRedirect("/labalive/auth/login/");
27         return;
28     }
29     try{
30         if (user.getToken() != null){
31             salutation = (user.getUsername() != null) ? "Hello " + user.↩
                   getUsername() + "! " : "";
32         }
33     }catch (Exception e){
34         response.sendRedirect("/labalive/auth/login/");
35     }
36
37     Collection<ResourceContainer> myApps = WiringSelector4Jsp.↩
          getAllWirings(request);
38
39
40 %>
41
42
43 <jsp:include page="/jsp/header.jsp" />
44 <section id="home_experiment">
45     <hr>
46     <h1>Resourcen</h1>
```

```
47 </section>
48
49 <!-------------------------------------------------->
50 <!-- my Apps ----------------------------------------->
51 <!-------------------------------------------------->
52 <section id="content">
53
54     <!-- Tab links -->
55     <div class="tab">
56         <button class="tablinks" onclick="openSection(event, 'About3')" ↩
               id="defaultOpen">Wiring</button>
57         <button class="tablinks" onclick="window.location.href='${↩
               pageContext.request.contextPath}/resource/signal.jsp';">↩
               Signal</button>
58         <button class="tablinks" onclick="window.location.href='${↩
               pageContext.request.contextPath}/resource/runWiring.jsp';">↩
               RunWiring</button>
59         <button class="tablinks" onclick="window.location.href='${↩
               pageContext.request.contextPath}/resource/queryString.jsp';"↩
               >QueryString</button>
60         <button class="tablinks" onclick="window.location.href='${↩
               pageContext.request.contextPath}/resource/layout.jsp';">↩
               Layout</button>
61         <button class="tablinks" onclick="window.location.href='${↩
               pageContext.request.contextPath}/resource/image.jsp';">Image↩
               </button>
62         <button class="tablinks" onclick="window.location.href='${↩
               pageContext.request.contextPath}/resource/experiment.jsp';">↩
               Experiment</button>
63     </div>
64
65
66     <!-------------------------------------------------->
67     <!-- tab --------------------------------------->
68     <!-------------------------------------------------->
69     <div id="About2" class="tabcontent">
70
71
72         <%
73             long resourceID = -1;
74             WiringResource selectedWiring = new WiringResource();
75
76             if (request.getParameter("resourceID") != null && Long.↩
                   parseLong(request.getParameter("resourceID")) != -1) {
77                 // existing wiring
78                 resourceID = Long.parseLong(request.getParameter("↩
```

```
                    resourceID"));
79
80                  for (ResourceContainer app : myApps) {
81                      WiringResource myWiring = (WiringResource) app.↩
                            getResource();
82                      if (myWiring.getResourceID() == resourceID) {
83                          selectedWiring = (WiringResource) app.↩
                                getResource();
84                      }
85                  }
86          %>
87          <p>
88              ResourceID:<%=resourceID%></p>
89          <br>
90
91          <%
92          } else {
93              selectedWiring.setTitle("New app");
94              selectedWiring.setClassname("mylab");
95              selectedWiring.setWiringDefinition("");
96              selectedWiring.setUserChanges("");
97          %>
98          <p>
99              WiringID:new</p>
100         <br>
101         <br>
102         <%
103             }
104             String wiringClassName = selectedWiring.getClassname();
105             String wiringVersion = Integer.toString(selectedWiring.↩
                    getVersion());
106         %>
107         <p>
108             WiringClassName:<%=wiringClassName%></p>
109         <br>
110
111     </div>
112
113
114     <!-------------------------------------------------->
115     <!-- Grid Layout ---------------------------------->
116     <!-------------------------------------------------->
117     <div id="About3" class="tabcontent">
118
119         <div class="grid-box">
120             <!-------------------------------------------------->
```

```
121             <!-- Grid Box Info ---------------------------------->
122             <!-------------------------------------------------->
123             <div class="box-info">
124                 <p><%= salutation %>Here are your apps:</p>
125             </div>
126 
127             <!-------------------------------------------------->
128             <!-- Grid Box Table ---------------------------------->
129             <!-------------------------------------------------->
130             <div class="box-table">
131                 <form method="get">
132                     <!--  <button onclick="document.submit()" href="/↩
                        labalive/my/apps/">New App</button> -->
133                     <label title="New app"><input type="submit" class="↩
                        submit-code" onclick="document.submit()" href="/↩
                        labalive/my/apps/"/><svg xmlns="http://www.w3.↩
                        org/2000/svg" viewBox="0 0 512 512" style="fill↩
                        :#494949; padding:0px 5px 0px 5px; height:20px; ↩
                        cursor:pointer;"><!--! Font Awesome Pro 6.2.1 by↩
                         @fontawesome - https://fontawesome.com License ↩
                        - https://fontawesome.com/license (Commercial ↩
                        License) Copyright 2022 Fonticons, Inc. --><path↩
                         d="M256 512c141.4 0 256-114.6 256-256S397.4 0 2↩
                        56 0S0 114.6 0 256S114.6 512 256 512zM232 344V28↩
                        0H168c-13.3 0-24-10.7-24-24s10.7-24 24-24h64V168↩
                        c0-13.3 10.7-24 24-24s24 10.7 24 24v64h64c13.3 0↩
                         24 10.7 24 24s-10.7 24-24 24H280v64c0 13.3-10.7↩
                         24-24 24s-24-10.7-24-24z"/></svg></label>
134                     <br>
135                     <br>
136                     <table id="myTable2" width="525px">
137                         <tr>
138                             <th onclick="sortTable(0)" width="25px">Id</↩
                                th>
139                             <th onclick="sortTable(1)">Title</th>
140                             <th onclick="sortTable(2)">Wiring Definition↩
                                </th>
141                             <!-- <th onclick="sortTable(3)">User Changes↩
                                </th>
142                             <th onclick="sortTable(4)">Version</th> -->
143                             <th onclick="sortTable(3)" width="85px">↩
                                Published Date</th>
144                             <th onclick="sortTable(4)" width="95px">↩
                                Options</th>
145                         </tr>
146                         <%
```

```
147                             for (ResourceContainer wiring : myApps){
148                                 //wiring.getResource();
149                                 String url = "/labalive/resource/wiring.↩
                                        jsp?resourceID=" + wiring.↩
                                        getResource().getResourceID();
150                                 WiringResource wiringResource = (↩
                                        WiringResource) wiring.getResource()↩
                                        ;
151
152
153                         %>
154                         <tr>
155                             <td><%=wiringResource.getResourceID()%></td>
156                             <td style="white-space: nowrap; overflow: ↩
                                    hidden; text-overflow: ellipsis; max-↩
                                    width: 14ch;"><%=wiringResource.getTitle↩
                                    ()%></td>
157                             <td style="white-space: nowrap; overflow: ↩
                                    hidden; text-overflow: ellipsis; max-↩
                                    width: 15ch;"><%=wiringResource.↩
                                    getWiringDefinition()%></td>
158                             <!-- <td><%=wiringResource.getUserChanges()↩
                                    %></td>
159                 <td><%=wiringResource.getVersion()%></td> -->
160                             <!-- <td></td>
161         <td></td> -->
162                             <td><%=wiringResource.getSaveDate()%></td>
163                             <!--        <td><a href="/labalive/webstart/↩
                                    mylab.jnlp"><img alt="Start" src="../../↩
                                    img/next.gif"></a><a onclick="document.↩
                                    submit()" href=""> Edit App</a></td>-->
164
165                             </td>
166                             <td>
167                                 <table class="noborder2">
168                                     <tr>
169                                         <td><a onclick="document.submit↩
                                                ()" href="<%=url%>" title="↩
                                                Edit App">
170                                          <svg xmlns="http://www.w3.↩
                                                org/2000/svg" viewBox="0↩
                                                 0 512 512" style="fill↩
                                                :#494949; height:20px; ↩
                                                cursor:pointer;"><!--! ↩
                                                Font Awesome Pro 6.2.0 ↩
                                                by @fontawesome - https:↩
```

```
                                        //fontawesome.com ↩
                                        License - https://↩
                                        fontawesome.com/license ↩
                                        (Commercial License) ↩
                                        Copyright 2022 Fonticons↩
                                        , Inc. --><path d="M471↩
                                        .6 21.7c-21.9-21.9-57.3-↩
                                        21.9-79.2 0L362.3 51.7l9↩
                                        7.9 97.9 30.1-30.1c21.9-↩
                                        21.9 21.9-57.3 0-79.2L47↩
                                        1.6 21.7zm-299.2 220c-6↩
                                        .1 6.1-10.8 13.6-13.5 21↩
                                        .9l-29.6 88.8c-2.9 8.6-↩
                                        .6 18.1 5.8 24.6s15.9 8↩
                                        .7 24.6 5.8l88.8-29.6c8↩
                                        .2-2.8 15.7-7.4 21.9-13↩
                                        .5L437.7 172.3 339.7 74↩
                                        .3 172.4 241.7zM96 64C43↩
                                         64 0 107 0 160V416c0 53↩
                                         43 96 96 96H352c53 0 96↩
                                        -43 96-96V320c0-17.7-14↩
                                        .3-32-32-32s-32 14.3-32 ↩
                                        32v96c0 17.7-14.3 32-32 ↩
                                        32H96c-17.7 0-32-14.3-32↩
                                        -32V160c0-17.7 14.3-32 3↩
                                        2-32h96c17.7 0 32-14.3 3↩
                                        2-32s-14.3-32-32-32H96z↩
                                        "/></svg>
171                                 </a></td>
172                                 <td><a onclick="document.submit↩
                                        ()" href="/labalive/wiring/↩
                                        delete/?resourceID=<%=↩
                                        wiringResource.getResourceID↩
                                        ()%>" title="Delete">
173                                      <svg xmlns="http://www.w3.↩
                                            org/2000/svg" viewBox="0↩
                                             0 448 512" style="fill↩
                                            :#494949; height:20px; ↩
                                            cursor:pointer;"><!--! ↩
                                            Font Awesome Pro 6.2.1 ↩
                                            by @fontawesome - https:↩
                                            //fontawesome.com ↩
                                            License - https://↩
                                            fontawesome.com/license ↩
                                            (Commercial License) ↩
                                            Copyright 2022 Fonticons↩
```

```
                                                , Inc. --><path d="M160 ↩
                                                400C160 408.8 152.8 416 ↩
                                                144 416C135.2 416 128 40↩
                                                8.8 128 400V192C128 183↩
                                                .2 135.2 176 144 176C152↩
                                                .8 176 160 183.2 160 192↩
                                                V400zM240 400C240 408.8 ↩
                                                232.8 416 224 416C215.2 ↩
                                                416 208 408.8 208 400V19↩
                                                2C208 183.2 215.2 176 22↩
                                                4 176C232.8 176 240 183↩
                                                .2 240 192V400zM320 400C↩
                                                320 408.8 312.8 416 304 ↩
                                                416C295.2 416 288 408.8 ↩
                                                288 400V192C288 183.2 29↩
                                                5.2 176 304 176C312.8 17↩
                                                6 320 183.2 320 192V400↩
                                                zM317.5 24.94L354.2 80H4↩
                                                24C437.3 80 448 90.75 44↩
                                                8 104C448 117.3 437.3 12↩
                                                8 424 128H416V432C416 47↩
                                                6.2 380.2 512 336 512H11↩
                                                2C67.82 512 32 476.2 32 ↩
                                                432V128H24C10.75 128 0 1↩
                                                17.3 0 104C0 90.75 10.75↩
                                                 80 24 80H93.82L130.5 24↩
                                                .94C140.9 9.357 158.4 0 ↩
                                                177.1 0H270.9C289.6 0 30↩
                                                7.1 9.358 317.5 24.94H31↩
                                                7.5zM151.5 80H296.5L277↩
                                                .5 51.56C276 49.34 273.5↩
                                                 48 270.9 48H177.1C174.5↩
                                                 48 171.1 49.34 170.5 51↩
                                                .56L151.5 80zM80 432C80 ↩
                                                449.7 94.33 464 112 464H↩
                                                336C353.7 464 368 449.7 ↩
                                                368 432V128H80V432z"/></↩
                                                svg>
                                        </a></td>
                                    </tr>
                                </table>
                            </td>
                        </tr>
                        <%
                            }
                        %>
```

```
182
183
184
185
186                     </table>
187                     <script>
188                         function sortTable(n) {
189                             var table, rows, switching, i, x, y, ↩
                                    shouldSwitch, dir, switchcount = 0;
190                             table = document.getElementById("myTable2");
191                             switching = true;
192                             // Set the sorting direction to ascending:
193                             dir = "asc";
194                             /* Make a loop that will continue until
195                             no switching has been done: */
196                             while (switching) {
197                                 // Start by saying: no switching is done↩
                                        :
198                                 switching = false;
199                                 rows = table.rows;
200                                 /* Loop through all table rows (except ↩
                                        the
201                                 first, which contains table headers): */
202                                 for (i = 1; i < (rows.length - 1); i++) ↩
                                        {
203                                     // Start by saying there should be ↩
                                            no switching:
204                                     shouldSwitch = false;
205                                     /* Get the two elements you want to ↩
                                            compare,
206                                     one from current row and one from ↩
                                            the next: */
207                                     x = rows[i].getElementsByTagName("TD↩
                                            ")[n];
208                                     y = rows[i + 1].getElementsByTagName↩
                                            ("TD")[n];
209                                     /* Check if the two rows should ↩
                                            switch place,
210                                     based on the direction, asc or desc:↩
                                             */
211                                     if (dir == "asc") {
212                                         if (x.innerHTML.toLowerCase() > ↩
                                                y.innerHTML.toLowerCase()) {
213                                             // If so, mark as a switch ↩
                                                    and break the loop:
214                                             shouldSwitch = true;
```

```
215                                         break;
216                                     }
217                                 } else if (dir == "desc") {
218                                     if (x.innerHTML.toLowerCase() < ↩
                                        y.innerHTML.toLowerCase()) {
219                                         // If so, mark as a switch ↩
                                            and break the loop:
220                                         shouldSwitch = true;
221                                         break;
222                                     }
223                                 }
224                             }
225                             if (shouldSwitch) {
226                                 /* If a switch has been marked, make↩
                                     the switch
227                                 and mark that a switch has been done↩
                                    : */
228                                 rows[i].parentNode.insertBefore(rows↩
                                    [i + 1], rows[i]);
229                                 switching = true;
230                                 // Each time a switch is done, ↩
                                    increase this count by 1:
231                                 switchcount ++;
232                             } else {
233                                 /* If no switching has been done AND↩
                                     the direction is "asc",
234                                 set the direction to "desc" and run ↩
                                    the while loop again. */
235                                 if (switchcount == 0 && dir == "asc"↩
                                    ) {
236                                     dir = "desc";
237                                     switching = true;
238                                 }
239                             }
240                         }
241                     }
242                 </script>
243                 <br>
244                 <!--      <button onclick="document.submit()" href="↩
                    /labalive/my/apps/">New App</button> -->
245                 <label title="New app"><input type="submit" class="↩
                    submit-code" onclick="document.submit()" href="/↩
                    labalive/my/apps/"/><svg xmlns="http://www.w3.↩
                    org/2000/svg" viewBox="0 0 512 512" style="fill↩
                    :#494949; padding:0px 5px 0px 5px; height:20px; ↩
                    cursor:pointer;"><!--! Font Awesome Pro 6.2.1 by↩
```

```
                        @fontawesome - https://fontawesome.com License ↩
                        - https://fontawesome.com/license (Commercial ↩
                        License) Copyright 2022 Fonticons, Inc. --><path↩
                         d="M256 512c141.4 0 256-114.6 256-256S397.4 0 2↩
                        56 0S0 114.6 0 256S114.6 512 256 512zM232 344V28↩
                        0H168c-13.3 0-24-10.7-24-24s10.7-24 24-24h64V168↩
                        c0-13.3 10.7-24 24-24s24 10.7 24 24v64h64c13.3 0↩
                         24 10.7 24 24s-10.7 24-24 24H280v64c0 13.3-10.7↩
                         24-24 24s-24-10.7-24-24z"/></svg></label>
246                 </form>
247             </div>
248
249             <!---------------------------------------------->
250             <!-- Grid Box Code ------------------------------->
251             <!---------------------------------------------->
252             <div class="box-code">
253                 <form method="get">
254
255                     <input type="hidden" name="resourceID" id="↩
                        resourceID" value="<%=resourceID%>">
256                     <input type="hidden" name="version" id="version" ↩
                        value="<%=wiringVersion%>">
257
258
259                     <table class="noborder">
260                         <tr>
261                             <td>App title:</td>
262                             <td>Class:</td>
263                         </tr>
264                         <tr>
265                             <td><textarea class="grid-ta" name="titel" ↩
                                id="titel" cols="50" rows="1"><%=↩
                                selectedWiring.getTitle()%></textarea></↩
                                td>
266                             <td><textarea class="grid-ta" name="↩
                                classname" id="classname" cols="50" rows↩
                                ="1"><%=selectedWiring.getClassname()↩
                                %></textarea></td>
267                         </tr>
268                         <tr>
269                             <td>Version:<%=selectedWiring.getVersion()↩
                                %></td>
270                             <td>Date:<%=selectedWiring.getSaveDate()%></↩
                                td>
271                         </tr>
272                     </table>
```

```
273                     <br>
274
275
276                     <table class="noborder">
277                         <tr>
278                             <th>Simulation description:</th>
279                         </tr>
280                         <tr>
281                             <td>
282                                 <textarea class="flex" name="wiringdef" ↩
                                        id="wiringdef" cols="100" rows="10"↩
                                        ><%=selectedWiring.↩
                                        getWiringDefinition()%></textarea>
283                             </td>
284                         </tr>
285                     </table>
286                     <br>
287                     <table class="noborder">
288                         <tr>
289                             <th>User changes:</th>
290                         </tr>
291                         <tr>
292                             <td>
293                                 <textarea class="flex" name="userchanges↩
                                        " id="userchanges" cols="100" rows="↩
                                        10"><%=selectedWiring.getUserChanges↩
                                        ()%></textarea>
294                             </td>
295                         </tr>
296                     </table>
297                     <table class="noborder">
298                         <tr>
299                             <td><label title="Edit Resource"><input id="↩
                                    editResource" type="submit" value="↩
                                    editResource" class="submit-code" ↩
                                    formaction="/labalive/resource/↩
                                    getresouceapp/" onclick="↩
                                    checkIfWiringIsSelected()" /><svg xmlns=↩
                                    "http://www.w3.org/2000/svg" viewBox="0 ↩
                                    0 512 512" style="fill:#494949; padding:↩
                                    0px 5px 0px 5px; height:20px; cursor:↩
                                    pointer;"> <path d="M224 256c-35.2 0-64 ↩
                                    28.8-64 64c0 35.2 28.8 64 64 64c35.2 0 6↩
                                    4-28.8 64-64C288 284.8 259.2 256 224 256↩
                                    zM433.1 129.1l-83.9-83.9C341.1 37.06 328↩
                                    .8 32 316.1 32H64C28.65 32 0 60.65 0 96v↩
```

```
                                320c0 35.35 28.65 64 64 64h320c35.35 0 6↩
                                4-28.65 64-64V163.9C448 151.2 442.9 138↩
                                .9 433.1 129.1zM128 80h144V160H128V80zM4↩
                                00 416c0 8.836-7.164 16-16 16H64c-8.836 ↩
                                0-16-7.164-16-16V96c0-8.838 7.164-16 16-↩
                                16h16v104c0 13.25 10.75 24 24 24h192C309↩
                                .3 208 320 197.3 320 184V83.88l78.25 78↩
                                .25C399.4 163.2 400 164.8 400 166.3V416z↩
                                "/></svg> </label></td>
300                         <script type="text/javascript">
301                             function checkIfWiringIsSelected(){
302                                 if (document.getElementById("↩
                                    resourceID").value<=0){
303                                     alert("Please Select a Wiring");
304                                     document.getElementById("↩
                                        editResource").disabled = ↩
                                        true;
305                                 }else {
306                                     document.getElementById("↩
                                        editResource").disabled = ↩
                                        false;
307                                 }
308 
309                             }
310                         </script>
311                         <td><label title="Launch"><input type="↩
                            submit" value="Launch" class="submit-↩
                            code" formaction="/labalive/webstart/↩
                            test.jnlp"/><img src="/labalive/img/jws-↩
                            launch-button.png" alt="Start" title="↩
                            Start via Java Web Start" onclick="gtag↩
                            ('event', 'editor.MyLab', {'↩
                            event_category': 'Launch', 'event_label↩
                            ': 'New app'});"></label></td>
312                         <td><label title="Download batch-file"><↩
                            input type="submit" value=".bat" class="↩
                            submit-code" formaction="/labalive/↩
                            batstart/test.bat"/><svg xmlns="http://↩
                            www.w3.org/2000/svg" viewBox="0 0 384 51↩
                            2" style="fill:#494949; padding:0px 5px ↩
                            0px 5px; height:20px; cursor:pointer;"↩
                            ><!--! Font Awesome Pro 6.2.1 by ↩
                            @fontawesome - https://fontawesome.com ↩
                            License - https://fontawesome.com/↩
                            license (Commercial License) Copyright 2↩
                            022 Fonticons, Inc. --><path d="M162.1 2↩
```

```
                                57.8c-7.812-7.812-20.47-7.812-28.28 0l-4↩
                                8 48c-7.812 7.812-7.812 20.5 0 28.31l48 ↩
                                48C137.8 386.1 142.9 388 148 388s10.23-1↩
                                .938 14.14-5.844c7.812-7.812 7.812-20.5 ↩
                                0-28.31L128.3 320l33.86-33.84C169.1 278↩
                                .3 169.1 265.7 162.1 257.8zM365.3 93.38l↩
                                -74.63-74.64C278.6 6.742 262.3 0 245.4 0↩
                                H64C28.65 0 0 28.65 0 64l.0065 384c0 35↩
                                .34 28.65 64 64 64H320c35.2 0 64-28.8 64↩
                                -64V138.6C384 121.7 377.3 105.4 365.3 93↩
                                .38zM336 448c0 8.836-7.164 16-16 16H64.0↩
                                2c-8.838 0-16-7.164-16-16L48 64.13c0-8.8↩
                                36 7.164-16 16-16h160L224 128c0 17.67 14↩
                                .33 32 32 32h79.1V448zM221.9 257.8c-7.81↩
                                2 7.812-7.812 20.5 0 28.31L255.7 320l-33↩
                                .86 33.84c-7.812 7.812-7.812 20.5 0 28.3↩
                                1C225.8 386.1 230.9 388 236 388s10.23-1↩
                                .938 14.14-5.844l48-48c7.812-7.812 7.812↩
                                -20.5 0-28.31l-48-48C242.3 250 229.7 250↩
                                 221.9 257.8z"/></svg></label></td>
313                         <td><div class="copyfunc">
314                             <label title="Copy2App"><input type="↩
                                submit" class="submit-code" onclick=↩
                                "myCopyFunction()" onmouseout="↩
                                outCopyFunc()"/><svg xmlns="http://↩
                                www.w3.org/2000/svg" viewBox="0 0 51↩
                                2 512" style="fill:#494949; padding:↩
                                0px 5px 0px 5px; height:20px; cursor↩
                                :pointer;"><!--! Font Awesome Pro 6↩
                                .2.1 by @fontawesome - https://↩
                                fontawesome.com License - https://↩
                                fontawesome.com/license (Commercial ↩
                                License) Copyright 2022 Fonticons, ↩
                                Inc. --><path d="M502.6 70.63l-61.25↩
                                -61.25C435.4 3.371 427.2 0 418.7 0H2↩
                                55.1c-35.35 0-64 28.66-64 64l.0195 2↩
                                56C192 355.4 220.7 384 256 384h192c3↩
                                5.2 0 64-28.8 64-64V93.25C512 84.77 ↩
                                508.6 76.63 502.6 70.63zM464 320c0 8↩
                                .836-7.164 16-16 16H255.1c-8.838 0-1↩
                                6-7.164-16-16L239.1 64.13c0-8.836 7↩
                                .164-16 16-16h128L384 96c0 17.67 14↩
                                .33 32 32 32h47.1V320zM272 448c0 8.8↩
                                36-7.164 16-16 16H63.1c-8.838 0-16-7↩
                                .164-16-16L47.98 192.1c0-8.836 7.164↩
                                -16 16-16H160V128H63.99c-35.35 0-64 ↩
```

```
                                    28.65-64 64l.0098 256C.002 483.3 28↩
                                    .66 512 64 512h192c35.2 0 64-28.8 64↩
                                    -64v-32h-47.1L272 448z"/></svg></↩
                                    label>
315                             </div>
316                             </td>
317                             <td>
318                                 <label title="Save"><input type="submit"↩
                                     name = "save" value="Save" class="↩
                                    submit-code" formaction="/labalive/↩
                                    wiring/createupdate/"/><svg xmlns="↩
                                    http://www.w3.org/2000/svg" viewBox=↩
                                    "0 0 448 512" style="fill:#494949; ↩
                                    padding:0px 5px 0px 5px; height:20px↩
                                    ; cursor:pointer;"><!--! Font ↩
                                    Awesome Pro 6.2.1 by @fontawesome - ↩
                                    https://fontawesome.com License - ↩
                                    https://fontawesome.com/license (↩
                                    Commercial License) Copyright 2022 ↩
                                    Fonticons, Inc. --><path d="M64 32C2↩
                                    8.7 32 0 60.7 0 96V416c0 35.3 28.7 6↩
                                    4 64 64H384c35.3 0 64-28.7 64-64V173↩
                                    .3c0-17-6.7-33.3-18.7-45.3L352 50.7C↩
                                    340 38.7 323.7 32 306.7 32H64zm0 96c↩
                                    0-17.7 14.3-32 32-32H288c17.7 0 32 1↩
                                    4.3 32 32v64c0 17.7-14.3 32-32 32H96↩
                                    c-17.7 0-32-14.3-32-32V128zM224 416c↩
                                    -35.3 0-64-28.7-64-64s28.7-64 64-64s↩
                                    64 28.7 64 64s-28.7 64-64 64z"/></↩
                                    svg></label>
319                             </td>
320                             <td><label title="Save as"><input type="↩
                                    submit" name ="save" value="Save as" ↩
                                    class="submit-code" formaction="/↩
                                    labalive/wiring/createupdate/"/><svg ↩
                                    xmlns="http://www.w3.org/2000/svg" ↩
                                    viewBox="0 0 448 512" style="fill:#49494↩
                                    9; padding:0px 5px 0px 5px; height:20px;↩
                                     cursor:pointer;"><!--! Font Awesome Pro↩
                                     6.2.1 by @fontawesome - https://↩
                                    fontawesome.com License - https://↩
                                    fontawesome.com/license (Commercial ↩
                                    License) Copyright 2022 Fonticons, Inc. ↩
                                    --><path d="M224 256c-35.2 0-64 28.8-64 ↩
                                    64c0 35.2 28.8 64 64 64c35.2 0 64-28.8 6↩
                                    4-64C288 284.8 259.2 256 224 256zM433.1 ↩
```

```
                                129.1l-83.9-83.9C341.1 37.06 328.8 32 31↩
                                6.1 32H64C28.65 32 0 60.65 0 96v320c0 35↩
                                .35 28.65 64 64 64h320c35.35 0 64-28.65 ↩
                                64-64V163.9C448 151.2 442.9 138.9 433.1 ↩
                                129.1zM128 80h144V160H128V80zM400 416c0 ↩
                                8.836-7.164 16-16 16H64c-8.836 0-16-7.16↩
                                4-16-16V96c0-8.838 7.164-16 16-16h16v104↩
                                c0 13.25 10.75 24 24 24h192C309.3 208 32↩
                                0 197.3 320 184V83.88178.25 78.25C399.4 ↩
                                163.2 400 164.8 400 166.3V416z"/></svg↩
                                ></label></td>
321                         <td><label title="Create link"><input type="↩
                                submit" value="Create link" class="↩
                                submit-code" formaction="/labalive/my/↩
                                apps/createlaunchbuttons/"/><svg xmlns="↩
                                http://www.w3.org/2000/svg" viewBox="0 0↩
                                 640 512" style="fill:#494949; padding:0↩
                                px 5px 0px 5px; height:20px; cursor:↩
                                pointer;"><!--! Font Awesome Pro 6.2.1 ↩
                                by @fontawesome - https://fontawesome.↩
                                com License - https://fontawesome.com/↩
                                license (Commercial License) Copyright 2↩
                                022 Fonticons, Inc. --><path d="M579.8 2↩
                                67.7c56.5-56.5 56.5-148 0-204.5c-50-50-1↩
                                28.8-56.5-186.3-15.4l-1.6 1.1c-14.4 10.3↩
                                -17.7 30.3-7.4 44.6s30.3 17.7 44.6 7.4l1↩
                                .6-1.1c32.1-22.9 76-19.3 103.8 8.6c31.5 ↩
                                31.5 31.5 82.5 0 114L422.3 334.8c-31.5 3↩
                                1.5-82.5 31.5-114 0c-27.9-27.9-31.5-71.8↩
                                -8.6-103.8l1.1-1.6c10.3-14.4 6.9-34.4-7↩
                                .4-44.6s-34.4-6.9-44.6 7.4l-1.1 1.6C206↩
                                .5 251.2 213 330 263 380c56.5 56.5 148 5↩
                                6.5 204.5 0L579.8 267.7zM60.2 244.3c-56↩
                                .5 56.5-56.5 148 0 204.5c50 50 128.8 56↩
                                .5 186.3 15.4l1.6-1.1c14.4-10.3 17.7-30↩
                                .3 7.4-44.6s-30.3-17.7-44.6-7.4l-1.6 1.1↩
                                c-32.1 22.9-76 19.3-103.8-8.6C74 372 74 ↩
                                321 105.5 289.5L217.7 177.2c31.5-31.5 82↩
                                .5-31.5 114 0c27.9 27.9 31.5 71.8 8.6 10↩
                                3.9l-1.1 1.6c-10.3 14.4-6.9 34.4 7.4 44↩
                                .6s34.4 6.9 44.6-7.4l1.1-1.6C433.5 260.8↩
                                 427 182 377 132c-56.5-56.5-148-56.5-204↩
                                .5 0L60.2 244.3z"/></svg></label></td>
322                         <th>
323                             <input  placeholder="Paste app" name="r"↩
                                    id="r" onClick="this.select();" ↩
```

```
                                type="text" style="width:120px;">
324                         </th>
325                         <th>
326                             <div class="pastefunc">
327                                 <label title="Paste to website"><↩
                                        input type="submit" class="↩
                                        submit-code" onclick="↩
                                        myPasteFunction()" onmouseout="↩
                                        outPasteFunc()"/><svg xmlns="↩
                                        http://www.w3.org/2000/svg" ↩
                                        viewBox="0 0 512 512" style="↩
                                        fill:#494949; padding:0px 5px 0↩
                                        px 5px; height:20px; cursor:↩
                                        pointer;"><!--! Font Awesome Pro↩
                                         6.2.1 by @fontawesome - https:↩
                                        //fontawesome.com License - ↩
                                        https://fontawesome.com/license ↩
                                        (Commercial License) Copyright 2↩
                                        022 Fonticons, Inc. --><path d="↩
                                        M128 64c0-35.3 28.7-64 64-64H352↩
                                        V128c0 17.7 14.3 32 32 32H512V44↩
                                        8c0 35.3-28.7 64-64 64H192c-35.3↩
                                         0-64-28.7-64-64V336H302.1l-39 3↩
                                        9c-9.4 9.4-9.4 24.6 0 33.9s24.6 ↩
                                        9.4 33.9 0l80-80c9.4-9.4 9.4-24↩
                                        .6 0-33.9l-80-80c-9.4-9.4-24.6-9↩
                                        .4-33.9 0s-9.4 24.6 0 33.9l39 39↩
                                        H128V64zm0 224v48H24c-13.3 0-24-↩
                                        10.7-24-24s10.7-24 24-24H128zM51↩
                                        2 128H384V0L512 128z"/></svg></↩
                                        label>
328                             </div>
329                         </th>
330                         <td><div class="usercopy">
331                             <label title="Copy user"><input type="↩
                                    submit" class="submit-code" onclick=↩
                                    "myUserFunction()" onmouseout="↩
                                    outUserFunc()"/><svg xmlns="http://↩
                                    www.w3.org/2000/svg" viewBox="0 0 44↩
                                    8 512" style="fill:#494949; padding:↩
                                    0px 5px 0px 5px; height:20px; cursor↩
                                    :pointer;"><!--! Font Awesome Pro 6↩
                                    .2.1 by @fontawesome - https://↩
                                    fontawesome.com License - https://↩
                                    fontawesome.com/license (Commercial ↩
                                    License) Copyright 2022 Fonticons, ↩
```

```
                                Inc. --><path d="M272 304h-96C78.8 3↩
                                04 0 382.8 0 480c0 17.67 14.33 32 32↩
                                 32h384c17.67 0 32-14.33 32-32C448 3↩
                                82.8 369.2 304 272 304zM48.99 464C56↩
                                .89 400.9 110.8 352 176 352h96c65.16↩
                                 0 119.1 48.95 127 112H48.99zM224 25↩
                                6c70.69 0 128-57.31 128-128c0-70.69-↩
                                57.31-128-128-128S96 57.31 96 128C96↩
                                 198.7 153.3 256 224 256zM224 48c44↩
                                .11 0 80 35.89 80 80c0 44.11-35.89 8↩
                                0-80 80S144 172.1 144 128C144 83.89 ↩
                                179.9 48 224 48z"/></svg></label>
332                         </div>
333                         </td>
334                         <td><label title="Delete"><input type="↩
                                submit" name = "delete" value="Delete" ↩
                                class="submit-code" formaction="/↩
                                labalive/wiring/delete/"/><svg xmlns="↩
                                http://www.w3.org/2000/svg" viewBox="0 0↩
                                 448 512" style="fill:#494949; padding:0↩
                                px 5px 0px 5px; height:20px; cursor:↩
                                pointer;"><!--! Font Awesome Pro 6.2.1 ↩
                                by @fontawesome - https://fontawesome.↩
                                com License - https://fontawesome.com/↩
                                license (Commercial License) Copyright 2↩
                                022 Fonticons, Inc. --><path d="M160 400↩
                                C160 408.8 152.8 416 144 416C135.2 416 1↩
                                28 408.8 128 400V192C128 183.2 135.2 176↩
                                 144 176C152.8 176 160 183.2 160 192V400↩
                                zM240 400C240 408.8 232.8 416 224 416C21↩
                                5.2 416 208 408.8 208 400V192C208 183.2 ↩
                                215.2 176 224 176C232.8 176 240 183.2 24↩
                                0 192V400zM320 400C320 408.8 312.8 416 3↩
                                04 416C295.2 416 288 408.8 288 400V192C2↩
                                88 183.2 295.2 176 304 176C312.8 176 320↩
                                 183.2 320 192V400zM317.5 24.94L354.2 80↩
                                H424C437.3 80 448 90.75 448 104C448 117↩
                                .3 437.3 128 424 128H416V432C416 476.2 3↩
                                80.2 512 336 512H112C67.82 512 32 476.2 ↩
                                32 432V128H24C10.75 128 0 117.3 0 104C0 ↩
                                90.75 10.75 80 24 80H93.82L130.5 24.94C1↩
                                40.9 9.357 158.4 0 177.1 0H270.9C289.6 0↩
                                 307.1 9.358 317.5 24.94H317.5zM151.5 80↩
                                H296.5L277.5 51.56C276 49.34 273.5 48 27↩
                                0.9 48H177.1C174.5 48 171.1 49.34 170.5 ↩
                                51.56L151.5 80zM80 432C80 449.7 94.33 46↩
```

```
                                4 112 464H336C353.7 464 368 449.7 368 43↩
                                2V128H80V432z"/></svg></label></td>
335                         <td><input type="hidden" value="pub" name="↩
                                pub">
336                          <input type="checkbox" name="pub" value=↩
                                 "pub" id="pub">Do you want to ↩
                                 publish your experiment?
337                         </td>
338                     </tr>
339                 </table>
340                 <br>
341             </form>
342             <br>
343
344
345
346         </div>
347         <script>
348
349             function myCopyFunction() {
350                 var id = document.getElementById("id");
351                 var t = document.getElementById("t");
352                 var c = document.getElementById("c");
353                 var w = document.getElementById("w");
354                 var p = document.getElementById("p");
355                 var user = '232323232323232'
356                 var checkbox = document.getElementById("pub");
357                 var version = document.getElementById("v");
358                 var pub = checkbox.checked;
359
360                 var copy = "*labAlive-app*\r\n"
361                     + "de.labAlive.wiring."
362                     + encodeURIComponent(c.value) + "\r\n"
363                     + "id=" + encodeURIComponent(id.value)
364                     + "&t=" + encodeURIComponent(t.value)
365                     + "&c=" + encodeURIComponent(c.value)
366                     + "&w=" + encodeURIComponent(w.value)
367                     + "&p=" + encodeURIComponent(p.value)
368                     + "&pub=" + encodeURIComponent(pub)
369                     + "&u=" + encodeURIComponent(user)
370                     + "&v=" + encodeURIComponent(v.value);
371
372                 navigator.clipboard.writeText(copy);
373
374                 event.preventDefault();
375
```

```
376                     var copyfunc = document.getElementById("cf");
377                     copyfunc.innerHTML = "Copied succsessful";
378                 }
379 
380                 var user = "232323232323232";
381                 function copy2Clipboard(className, parameters) {
382                     var copy = "*labAlive-app*\r\n"
383                         + "de.labAlive.wiring."
384                         + encodeURIComponent(className) + "\r\n"
385                         + parameters
386                         + "&u=" + encodeURIComponent(user);
387 
388                     navigator.clipboard.writeText(copy);
389 
390                     var copyfunc = document.getElementById("cf");
391                     copyfunc.innerHTML = "Copied succsessful";
392                 }
393 
394                 function outCopyFunc() {
395                     var copyfunc = document.getElementById("cf");
396                     copyfunc.innerHTML = "Copy to clipboard";
397                 }
398 
399                 function myPasteFunction() {
400                     var paste = document.getElementById("r").value;
401                     var tmp = paste.split(" ");
402                     var check = tmp[0];
403                     event.preventDefault();
404                     if(check == "*labAlive-app*"){
405                         var params = tmp[2].split("&");
406                         for(i = 0; i<params.length; i++){
407                             var subparam = params[i].split("=");
408                             switch(subparam[0]) {
409                                 case "u":
410                                     break;
411                                 case "pub":
412                                     break;
413                                 case "v":
414                                     document.getElementById("version").↩
                                        value = decodeURIComponent(↩
                                        subparam[1]);
415                                     break;
416                                 default:
417                                     document.getElementById(subparam[0])↩
                                        .value = decodeURIComponent(↩
                                        subparam[1]);
```

```
418                             }
419
420                         }
421                     }
422                     document.getElementById("r").value = " ";
423                     var pastefunc = document.getElementById("pf");
424                     pastefunc.innerHTML = "Copied succsessful";
425                 }
426
427                 function outPasteFunc() {
428                     var pastefunc = document.getElementById("pf");
429                     pastefunc.innerHTML = "Paste to website";
430                 }
431
432                 function myUserFunction() {
433                     var user = "*labAlive-user*\r\nu=" + '23232323232323↩
                        2'
434                     navigator.clipboard.writeText(user);
435
436                     var usercopy = document.getElementById("uc");
437                     usercopy.innerHTML = "Copied succsessful";
438                 }
439                 function outUserFunc() {
440                     var usercopy = document.getElementById("uc");
441                     usercopy.innerHTML = "Copy user";
442                 }
443
444             </script>
445         </div>
446     </div>
447 </section>
448 <jsp:include page="/jsp/footer.jsp" />
```

Quellcode 6.7: GUI Wiring

# Stichwortverzeichnis

# Literaturverzeichnis


[1] *Microsoft Visio*. Microsoft-Visio.
URL: https://de.wikipedia.org/wiki/Microsoft_Visio (besucht am 26. 06. 2023) (siehe Seite 3).

[2] *Gitea*. Gitea. URL: https://de.wikipedia.org/wiki/Gitea (besucht am 26. 06. 2023) (siehe Seite 3).

[3] *IntelliJ*. IntelliJ. URL: https://de.wikipedia.org/wiki/IntelliJ_IDEA (besucht am 26. 06. 2023) (siehe Seite 4).

[4] *Microsoft-Access*. Microsoft-Access.
URL: https://de.wikipedia.org/wiki/Microsoft_Access (besucht am 26. 06. 2023) (siehe Seite 4).

[5] *Burp-Suite-Community-Edition*. Burp-Suite-Community-Edition.
URL: https://portswigger.net/burp/communitydownload (besucht am 26. 06. 2023) (siehe Seite 4).

[6] *Java*. Java.
URL: https://de.wikipedia.org/wiki/Java_(Programmiersprache) (besucht am 26. 06. 2023) (siehe Seite 5).

[7] *JavaScript*. JavaScript. URL: https://de.wikipedia.org/wiki/JavaScript (besucht am 26. 06. 2023) (siehe Seite 5).

[8] *HTML*. HTML.
URL: https://de.wikipedia.org/wiki/Hypertext_Markup_Language (besucht am 26. 06. 2023) (siehe Seite 5).

[9] *SQL*. SQL. URL: https://de.wikipedia.org/wiki/SQL (besucht am 26. 06. 2023) (siehe Seite 5).

[10] *SQL-Injection*. SQL Injection.
URL: https://owasp.org/www-community/attacks/SQL_Injection (besucht am 26. 06. 2023) (siehe Seite 22).

[11] *Retrieving hidden data*. Retrieving hidden data.
URL: https://portswigger.net/web-security/sql-injection#retrieving-hidden-data (besucht am 26. 06. 2023) (siehe Seite 24).

[12] *UNION-Angriffe*. UNION-Angriffe.
URL: https://portswigger.net/web-security/sql-injection/union-attacks (besucht am 26. 06. 2023) (siehe Seite 24).

[13] *Examining the database*. Examining the database.
URL: https://portswigger.net/web-security/sql-injection/examining-the-database (besucht am 26. 06. 2023) (siehe Seite 25).

[14] *Blind SQL injection*. Blind SQL injection.
URL: https://portswigger.net/web-security/sql-injection/blind (besucht am 26. 06. 2023)
(siehe Seite 25).

[15] *How to prevent SQL injection*. How to prevent SQL injection.
URL: https://portswigger.net/web-security/sql-injection#how-to-prevent-sql-injection (besucht am 26. 06. 2023) (siehe Seite 26).

[16] *Cross-Site-Scripting*. Cross-Site-Scripting.
URL: https://portswigger.net/web-security/cross-site-scripting (besucht am 26. 06. 2023)
(siehe Seite 26).

[17] *Reflected-XSS*. Reflected-XSS.
URL: https://portswigger.net/web-security/cross-site-scripting/reflected (besucht am 26. 06. 2023)
(siehe Seite 27).

[18] *Stored.XSS*. Stored.XSS.
URL: https://portswigger.net/web-security/cross-site-scripting/stored (besucht am 26. 06. 2023)
(siehe Seite 27).

[19] *How to prevent XSS*. How to prevent XSS.
URL: https://portswigger.net/web-security/cross-site-scripting/preventing (besucht am 26. 06. 2023)
(siehe Seite 27).