

Studienarbeit

Recherche, Bewertung, Implementierung und automatisiertes Testen von Groupware-Systemen

im Studiengang Softwaretechnik und Medieninformatik (SWB)
der Fakultät Informationstechnik
Wintersemester 2023-2024

Kyle Mezger
Matrikelnummer: 765838

Datum: 15.02.2023

Erstprüfer: Prof. Dr. -Ing. Andreas Röbler

Zweitprüfer: Prof. Dr. rer. nat. Jörg Nitzsche

Eidesstattliche Erklärung

Hiermit versichere ich, Kyle Mezger, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Weiterhin erkläre ich, dass ich für die Umformulierung einzelner Textpassagen und die Korrektur von Rechtschreib- und Grammatikfehlern die Hilfe von digitalen Tools, speziell ChatGPT und DeepL Write, in Anspruch genommen habe. Diese Unterstützung betraf ausschließlich die Formulierungshilfe und die Korrektur von Grammatik und Rechtschreibung, ohne dass die inhaltliche Eigenständigkeit und Urheberschaft meiner Arbeit dadurch berührt wurden.

Esslingen, den 23. Februar 2024

Unterschrift

Inhaltsverzeichnis

1	Kurzfassung	5
2	Einleitung	6
2.1	Motivation für die Studienarbeit	6
2.2	Ziele der Studienarbeit	6
3	Grundlagen	8
3.1	Groupware-Systeme	8
3.2	Playwright	10
4	Recherche und Installation des Groupware-Systems	12
4.1	Kriterien für das Groupware-System	12
4.2	Betrachtete Groupware-Systeme	13
4.2.1	Kolab	13
4.2.2	Horde	13
4.2.3	Sogo	14
4.2.4	EGroupware	14
4.3	Entscheidung für ein Groupware-System	15
4.4	Installation auf der Cloud	15
5	Testing der Groupware	17
5.1	Aufsetzen der Testumgebung	17
5.2	Implementierung der Tests	19
5.3	Ausführen der Tests	22
6	Zusammenfassung	24
6.1	Fazit	24
6.2	Ausblick	24

Abbildungsverzeichnis

1	Screenshot von Microsoft Outlook Live Mail	8
2	Screenshot von Microsoft Outlook Live Kalender	9
3	Screenshot von Microsoft Outlook Live Kontakte	10
4	Konfiguration der Sicherheitsgruppe für die bwCloud-Instanz von EGroup- ware	16
5	Beispiel für das Objekt des Administrator-Accounts	17
6	Playwright Codegen generierter Code für das Ausfüllen eines Login-Formulars	18
7	Playwright Pick-Locator Funktion zum Generieren von Selektoren für UI- Elemente	19
8	Login Test	20
9	Objekt zum Erstellen eines Termins innerhalb eines Tests	20
10	Datepicker UI-Element in EGroupware	21
11	Beispiel für das Objekt zum Erstellen eines Testnutzers	22
12	Testergebnisse der Playwright-Tests	23

1 Kurzfassung

Diese Arbeit beschäftigt sich mit der Recherche, Evaluierung, Implementierung und dem automatisierten End-to-End-Testing von Groupware-Systemen für die Hochschule Esslingen. Besonderer Wert wird auf den Einsatz von Open-Source-Software gelegt, die selbst administrierbar sein soll und möglichst von einer deutschen Firma entwickelt wurde.

Groupware-Systeme sind Softwareanwendungen, die die Zusammenarbeit und Organisation von Arbeitsgruppen unterstützen. Sie bieten Funktionen wie das Anlegen von gemeinsamen Terminen, das Erstellen von Projektplänen sowie das Versenden und Empfangen von E-Mails. In dieser Studienarbeit wird der Prozess der Recherche und Bewertung verschiedener Groupware-Systeme erläutert.

Im Umfang der Recherche werden vier verschiedene Groupware-Systeme und deren Funktionen beschrieben und anhand der gegebenen Kriterien bewertet. Es wird begründet, warum das Groupware-System EGroupware ausgewählt wurde und wie das System auf einer Cloud-Instanz installiert und konfiguriert wurde.

Im Anschluss geht die Studienarbeit auf den Aufbau der Testumgebung und die Implementierung der Tests ein, wobei auch die Arbeitsweise mit der Testing-Bibliothek Playwright und dessen Tools beschrieben wird. Zudem wird genauer auf die technische Implementierung verschiedener automatisierter End-to-End-Tests für EGroupware eingegangen, die alltägliche Nutzerszenarien abdecken sollen. Dabei werden wichtige Konzepte für die Implementierung und die abgedeckten Bereiche der Tests näher erläutert.

Abschließend wird ein Fazit über die Ergebnisse der Studienarbeit gezogen. Es wird bewertet, ob EGroupware als Groupware-System für die Hochschule Esslingen geeignet ist. Zudem wird ein Ausblick auf mögliche nächste Schritte in der Suche nach einer Groupware-Lösung für die Hochschule Esslingen gegeben. Dazu gehört die Untersuchung weiterer Groupware-Systeme aus der Recherche dieser Arbeit in weiteren Studienarbeiten sowie das ausführlichere Testen von EGroupware mit leistungsfähigerer System-Hardware und anspruchsvolleren Test-Cases.

2 Einleitung

In diesem Kapitel wird auf die Motivation für die Suche eines neuen Groupware-Systems für die Hochschule Esslingen eingegangen und die Ziele der genaueren Untersuchung eines der Groupware-Systeme erläutert.

2.1 Motivation für die Studienarbeit

Die Hochschule Esslingen setzt momentan Microsoft Exchange als Groupware-System ein. Der aktuelle Vertrag der Hochschule mit Microsoft läuft jedoch im Jahr 2024 aus. Microsoft gibt jedoch noch keine Informationen über die zukünftigen datenschutztechnischen Eigenschaften von Microsoft Exchange. Aufgrund dessen und weiterer datenschutzrechtlicher Bedenken mit Microsoft Exchange, soll bereits vor Ablauf des Vertrags ein neues Groupware-System als Alternative für Microsoft Exchange gefunden werden. Um bei der Entscheidung zu helfen, soll diese Studienarbeit einen ersten Überblick über verfügbare Open-Source-Groupware-Systeme geben und mindestens ein Groupware-System auf seine Eignung für die Hochschule Esslingen mit End-to-End-Tests überprüfen.

Aufbauend auf diese Studienarbeit sollen dann eventuell weitere Studienarbeiten folgen, die weitere Groupware-Systeme testen und bewerten. So soll der Hochschule Esslingen eine fundierte Entscheidungshilfe für die Auswahl eines neuen Groupware-Systems als potenzielle Alternative zu Microsoft Exchange gegeben werden.

2.2 Ziele der Studienarbeit

Im Laufe der Studienarbeit sollen verschiedene Open-Source-Groupware-Systeme recherchiert und basierend auf ihre Eignung als Groupware-System für die Hochschule Esslingen bewertet werden. Davon soll mindestens ein Groupware-System anhand später erläuterter Kriterien ausgewählt und anschließend in einer Testumgebung installiert und grundlegend konfiguriert werden. Dieses System soll dann automatisiert mit der User-Interface-Testing-Bibliothek Playwright getestet werden. Die Playwright-Tests sollen übliche Anwendungsfälle von Nutzern und Administratoren abdecken und eine erste Einschätzung über die Qualität bzw. Stabilität des Systems geben.

Über das getestete Groupware-System soll dann anhand der Tests ein Fazit gezogen werden, ob es für die Anforderungen Hochschule Esslingen geeignet ist. Dabei soll auch ein Ausblick auf mögliche nächste Schritte in der Suche nach einer Groupware-Lösung für die Hochschule Esslingen gegeben werden.

3 Grundlagen

Dieses Kapitel soll technologische und konzeptionelle Grundlagen für das Verständnis der Studienarbeit liefern. Zuerst werden anhand eines Beispiels Groupware-Systeme und deren grundlegende Funktionalitäten dargestellt, um ein Verständnis für die Funktionen von Groupware-Systemen zu schaffen. Daraufhin wird die Open-Source-Bibliothek Playwright vorgestellt, die in dieser Studienarbeit für die Implementierung von automatisierten End-to-End-Tests verwendet wurde. Dabei wird auf die Gründe und Vorteile von End-to-End-Tests, sowie die Funktionsweise von Playwright eingegangen.

3.1 Groupware-Systeme

Groupware-Systeme sind Software-Systeme, welche die Zusammenarbeit von Benutzern mit verschiedenen Tools zur gemeinsamen Kommunikation und Organisation unterstützen. Dabei werden üblicherweise Funktionalitäten wie Kalender, Terminplanung, E-Mails und Kontaktmanagement bereitgestellt.

Im Folgenden werden anhand von Screenshots aus Microsoft Outlook Live einige essenzielle Funktionen von Groupware-Systemen dargestellt und erklärt.

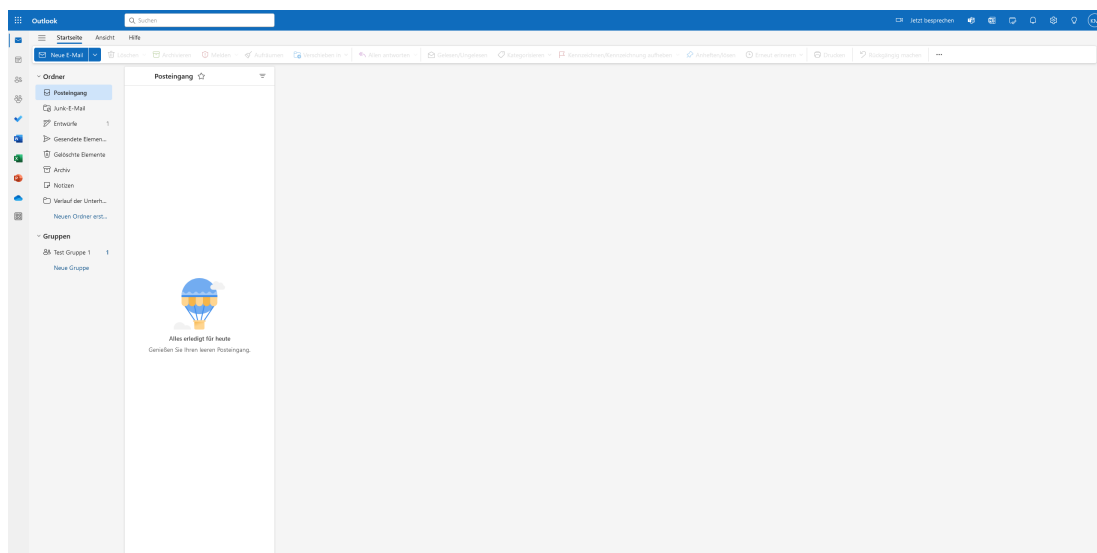


Abbildung 1: Screenshot von Microsoft Outlook Live Mail

Quelle: <https://outlook.live.com/calendar/0/view/month>, abgerufen am 29.01.2024

Die erste Hauptfunktion von Groupware-Systemen besteht darin, einen E-Mail-Client bereitzustellen, über den E-Mails empfangen, versendet und verwaltet werden können. Dies wird beispielhaft anhand von Outlook Live in Abbildung 1 dargestellt. Zudem sollten alle Funktionen eines modernen E-Mail-Clients, wie beispielsweise das Sortieren von E-Mails in Ordnern oder das Hinzufügen mehrerer E-Mail-Postfächer gleichzeitig.

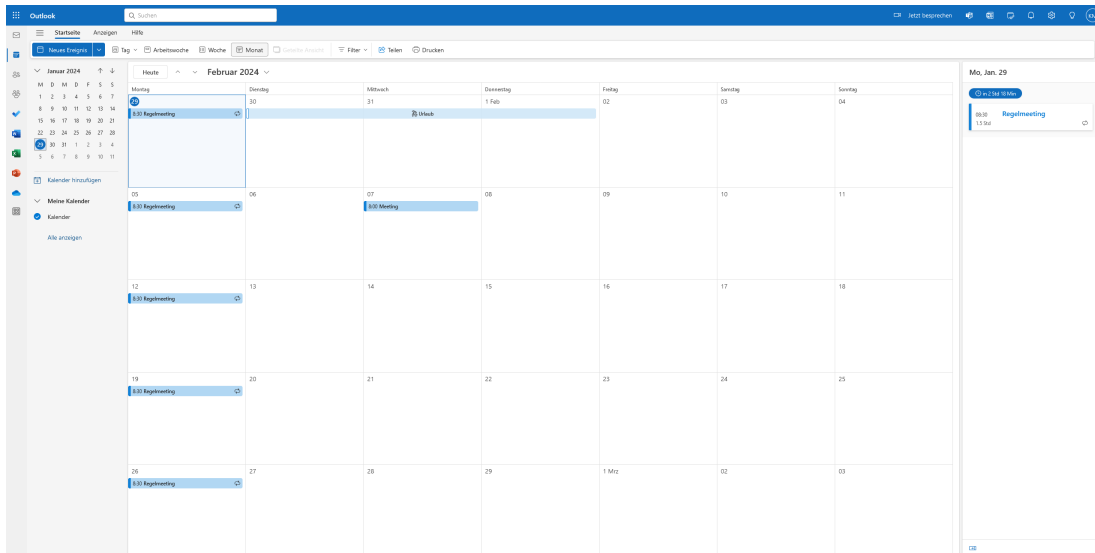


Abbildung 2: Screenshot von Microsoft Outlook Live Kalender

Quelle: <https://outlook.live.com/mail/0/>, abgerufen am 29.01.2024

Eine weitere Hauptfunktion von Groupware-Systemen ist ein Kalender, der Nutzern eine Terminplanung mit Ereignissen ermöglicht (siehe Abbildung 2). Die Terminplanung sollte es den Nutzern ermöglichen, andere Nutzer aus den Kontakten einzuladen, um die Zusammenarbeit und gemeinsame Organisation zu vereinfachen. Auch das Anlegen von Regelterminen, das heißt Terminen, die in regelmäßigen Zeitabständen wiederkehrend sind, sollte möglich sein.

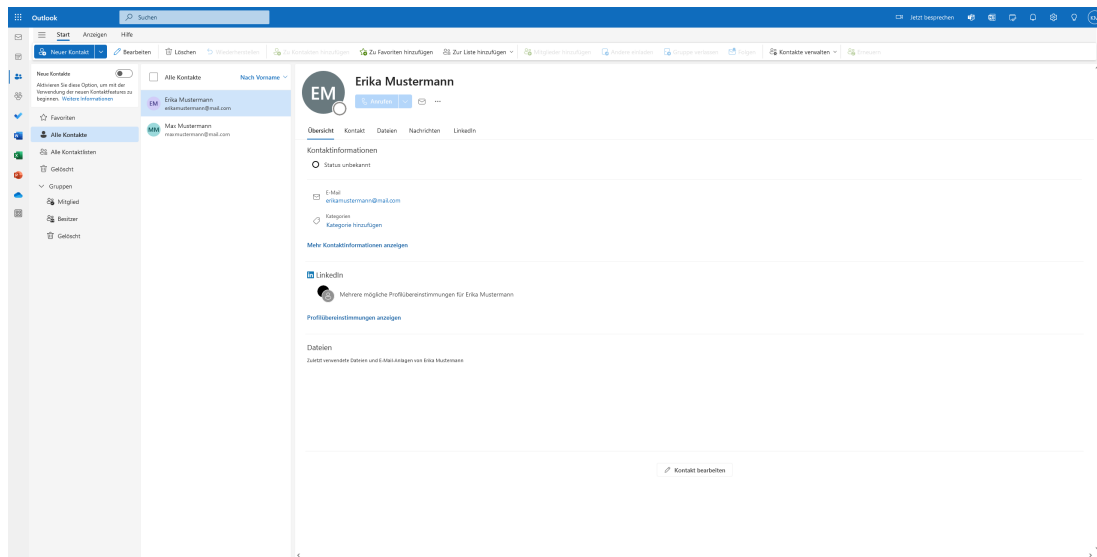


Abbildung 3: Screenshot von Microsoft Outlook Live Kontakte

Quelle: <https://outlook.live.com/people/0/>, abgerufen am 29.01.2024

Kontakte sind ein weiterer wichtiger Bestandteil von Groupware-Systemen, um die Vernetzung innerhalb von Arbeitsgruppen zu organisieren. Sie ermöglichen eine einfache Kontaktaufnahme zu anderen Gruppenmitgliedern. In Outlook-Live kann man, wie in Abbildung 3 gezeigt, direkt vom Kontakt einer Person diese Person per Nachricht, Anruf oder E-Mail kontaktieren.

3.2 Playwright

Die Open-Source-Bibliothek Playwright, die Anfang 2020 von Microsoft veröffentlicht wurde, ermöglicht die gesteuerte Automatisierung von Browsern und Webinterfaces. Dadurch ist es möglich, Webanwendungen automatisiert zu testen oder Websites abzutasten. Dabei bietet Playwright ein Application-Programming-Interface (API) für die Programmiersprachen JavaScript, TypeScript, Python, .NET und Java, sowie eine Vielzahl von Funktionen, die das Testen von Webanwendungen erleichtern. Beispielsweise kann mit Playwright-Codegen die eigene Interaktion mit einer Webanwendung aufgezeichnet und als Code exportiert werden, der dann als Test für die ausgeführte Interaktion verwendet werden kann. So können effizient Frontend-Tests für eine Vielzahl von Anwendungen implementiert werden (1).

Im Fall der Studienarbeit wurde Playwright verwendet, um automatisierte End-to-End-Tests für das final ausgewählte Groupware-System durchzuführen. End-to-End-Tests sind Tests, die reale Nutzerszenarien simulieren, um sicherzustellen, dass die Anwendung wie gewünscht funktioniert. Die Tests sind dabei so konzipiert, dass sie die Anwendung aus

Sicht eines Endnutzers testen und so das korrekte Betriebsverhalten aus einer realen Umgebung sicherstellen. So können sie, vor allem bei Anwendungen, die viele Nutzerinteraktionen erfordern, die User-Experience verbessern (vgl. 2). Dafür werden Frontend-Tests implementiert, die typische Interaktionen mit der Benutzeroberfläche simulieren. Beispielsweise können so Formulare ausgefüllt oder Buttons angeklickt werden, womit beispielsweise ein Nutzer-Login und das anschließende Aufrufen der Mails des Nutzers simuliert werden kann.

Deckt man mit diesen Tests alle Funktionsbereiche des Groupware-Systems ab, kann man durch das Ausführen der Tests sicherstellen, dass die vollständige Anwendung nach einer Änderung noch wie erwartet funktioniert. Auch falls die Anwendung in Zukunft unerwartete Ausfälle generiert, können diese durch flächendeckende Tests genauer erkannt werden, da sofort ersichtlich ist, welche Bereiche des Systems noch funktionieren und welche nicht. Geht beispielsweise der zuvor erwähnte Test des aufrufen der Mails schief, gibt es mit hoher Wahrscheinlichkeit ein Problem mit der Verbindung zum Mail Server.

Zudem können solche Tests in Playwright in verschiedenen Browsern (Chromium, Firefox, WebKit) ausgeführt werden, wodurch die Funktionalität der Anwendung auch auf verschiedenen Browsern sichergestellt und kontinuierlich getestet werden kann. Da die Groupware von einer Vielzahl an Systemen zugänglich sein soll, ist diese Funktion ein essenzieller Bestandteil der Testanforderungen.

4 Recherche und Installation des Groupware-Systems

In diesem Kapitel werden der Prozess und die Ergebnisse der Recherche mehrerer Open-Source-Groupware-Systeme und die anschließende Installation eines der Systeme beschrieben. Dabei wird zunächst auf die Kriterien für das Groupware-System eingegangen, die im Laufe der Recherche berücksichtigt wurden. Anschließend werden die betrachteten Groupware-Systeme vorgestellt und die Entscheidung für eines der Systeme begründet. Zuletzt wird die Installation und Konfiguration des ausgewählten Groupware-Systems auf einer Cloud-Instanz beschrieben.

4.1 Kriterien für das Groupware-System

- **Open Source:** Das erste und wichtigste Kriterium ist, dass das Groupware-System Open-Source ist. Open-Source-Anwendungen sind frei verfügbar und können von der Hochschule Esslingen selbst installiert und administriert werden. Zudem sind Open-Source-Anwendungen bezüglich auf Datenschutz oft sicherer und transparenter, da der Quellcode von Jedem eingesehen und geprüft werden kann. Daher werden im Laufe der Recherche nur Open-Source-Groupware-Systeme betrachtet.
- **Eigenadministration durch die Hochschule:** Die Hochschule Esslingen hat ein eigenes Rechenzentrum und eine IT-Fakultät. Daher sollte die Software von der Hochschule Esslingen selbst installiert und administriert werden können.
- **Automatisierung:** Teile der Konfiguration des Groupware-Systems, wie beispielsweise das Hinzufügen und die Konfiguration von Nutzern, sollen automatisierbar sein, beziehungsweise über Skripte gelöst werden können. Das soll zum Beispiel das Hinzufügen und Entfernen der vielen Nutzer, die an der Hochschule Esslingen beschäftigt sind, erleichtern.
- **Deutsche Firma:** Als deutsche Hochschule möchte die Hochschule Esslingen auch deutsche Firmen unterstützen, weshalb die Groupware, wenn möglich, von einer deutschen Firma entwickelt sein soll. Eine Nichterfüllung dieses Kriteriums muss aber nicht zwingend zum Ausschluss führen, es wird eher als Bonus-Kriterium angesehen.

4.2 Betrachtete Groupware-Systeme

Zu Beginn der Studienarbeit wurden anhand der gegebenen Kriterien mehrere Kandidaten für das Groupware-System recherchiert, um einen Überblick über die verfügbaren Möglichkeiten zu erhalten. Im folgenden Abschnitt werden die 4 vielversprechendsten Groupware-Systeme der ersten Recherche, Kolab, Horde, Sogo und EGroupware, mit ihren Funktionalitäten und anderen relevanten Informationen vorgestellt. Dabei wird auch auf die Verbreitung der Systeme an anderen Organisationen, vor allem aber Hochschulen und Universitäten, eingegangen. Das soll eine erste Einschätzung über die Eignung der Systeme zur eigenen Verwaltung durch die Hochschule geben.

4.2.1 Kolab

Das Groupware-System Kolab wird von der Schweizer Firma Aphelia IT AG entwickelt und ist als Open-Source-Produkt gratis verfügbar. Es bietet die folgenden Features:

- E-Mail
- Kalender
- Kontakte
- Online-File-Server
- Aufgabenmanagement
- Notizen
- Sprach- und Videoanrufe

(3)

Kolab wird von der Firma Nestlé und der Universität Tübingen verwendet und ist daher vermutlich auch für größere Organisationen geeignet.

4.2.2 Horde

Horde wird von einem gleichnamigen amerikanischen Unternehmen, Horde LLC, entwickelt und ist wie die anderen Systeme Open-Source und gratis verfügbar. Es bietet dabei die folgenden Funktionalitäten:

- E-Mail
- Kalender
- Kontakte
- Online-File-Server

- Terminmanagement
- Projektmanagement
- Dokumentenmanagement

(4)

Horde wird schon von einigen anderen Universitäten und Hochschulen, wie beispielsweise der Universität Tübingen und Universität Paderborn als Webmail-Software, verwendet.

4.2.3 Sogo

Sogo ist ein Open-Source-Groupware-System, welches von der französischen Firma Alinto entwickelt wird. Es nutzt die Microsoft Exchange ActiveSync Technologie und kann daher auch in bestehenden Groupware-Systemen wie Microsoft Outlook oder Apple Kalender integriert werden. Das System ist frei verfügbar und bietet die folgenden Features:

- E-Mail
- Kalender
- Kontakte
- Erinnerungen
- 2-Faktor-Authentifizierung
- Raum Reservationen

(5)

Sogo wird von den untersuchten Groupware-Systemen von den meisten Universitäten und Hochschulen verwendet, wie beispielsweise der Universität Koblenz und der Universität Ulm.

4.2.4 EGroupware

EGroupware ist ein Open-Source-Groupware-System, welches von einer deutschen Firma entwickelt wird. Die Firma bietet eine gratis Community Edition an, welche folgende Funktionen bietet:

- E-Mail
- Kalender
- Kontakte
- Online-File-Server

- Terminmanagement
- Projektmanagement
- Dokumentenmanagement

Auch EGroupware wird von einigen Hochschulen genutzt, jedoch von deutlich weniger als die anderen recherchierten Systeme.

4.3 Entscheidung für ein Groupware-System

Bei der Recherche der verschiedenen Groupware-Systeme wurde klar, dass alle der untersuchten Systeme die grundlegenden gewünschten Funktionalitäten bieten und daher grundsätzlich alle für die Hochschule Esslingen geeignet sind. Durch diesen Umstand konnte die Entscheidung nicht rein aufgrund der Funktionalitäten der Systeme getroffen werden, da sich keines der Systeme in diesem Punkt stark von den anderen abhebt. Somit wurde final die Entscheidung basierend auf dem Bonus-Kriterium, dass das System von einer deutschen Firma entwickelt wird, getroffen.

4.4 Installation auf der Cloud

Für die Installation der EGroupware-Software stellt die EGroupware GmbH eine Installationsanleitung zu Verfügung. Darin wird auf einer Linux Instanz zunächst ein Docker Repository von EGroupware hinzugefügt und anschließend die EGroupware-Software in Form von 5 Docker Containern installiert und gestartet. Alle dafür benötigten Konsolenbefehle waren in der Installationsanleitung angegeben (6).

Für das Hosting der EGroupware-Software wurden sowohl WSL2 (Windows-Subsystem für Linux) als auch eine bwCloud-Instanz getestet. Die reine Installation der Software verlief auf beiden Systemen auf einer Ubuntu 22.04 Instanz ohne Probleme. Im Laufe der Konfiguration wurde jedoch klar, dass die Netzwerkkonfiguration bei bwCloud einfacher zu handhaben ist als bei WSL2. Daher wurden die finale Installation und Konfiguration nur auf einer bwCloud-Instanz durchgeführt. Damit das Frontend der Software über das Internet erreichbar ist, muss Port 80 für HTTP Kommunikation freigegeben werden, was, wie in Abbildung 4 gezeigt, in einer benutzerdefinierten Sicherheitsgruppe der bwCloud-Instanz realisiert wurde.

Manage Security Group Rules: eGroupware (4decc513-8922-47a3-bcf7-a172ec28906a)

+ Add Rule

Delete Rules

Displaying 1 item

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	HTTP	<div>Delete Rule</div>

Displaying 1 item

Abbildung 4: Konfiguration der Sicherheitsgruppe für die bwCloud-Instanz von EGroupware

Für die grundlegende Konfiguration der Software, wie beispielsweise das Konfigurieren von LDAP oder SAML 2.0, bietet die EGroupware Software ein Setup-Tool, welches über folgende URL erreichbar ist:

`http://host-or-ip/egroupware/setup/`

Für den ersten Zugriff auf die tatsächliche Software, beispielsweise für das Erstellen von Nutzer-Accounts, wird bei der Installation automatisch ein Administrator-Account angelegt, dessen Zugangsdaten im Protokoll der Installation zu finden sind. Das Frontend der Groupware ist dann über die folgende URL erreichbar:

`http://host-or-ip/egroupware/login.php/`

5 Testing der Groupware

Nach erfolgreicher Installation von EGroupware kann das System über jeden Browser aufgerufen werden und kann daher getestet werden. Dabei wird das System mithilfe von Playwright getestet. In diesem Kapitel wird dabei das Aufsetzen der Playwright Testumgebung und die Implementierung der Tests näher erläutert.

5.1 Aufsetzen der Testumgebung

Für das Entwickeln der Tests wird die Entwicklungsumgebung Visual Studio Code verwendet da es durch die Erweiterung „Playwright Test for VSCode“ von Microsoft eine sehr gute Integration der Playwright API bietet. Mit dieser Erweiterung kann auch die vollständige Installation von Playwright in dem aktuellen Projektordner direkt in der Entwicklungsumgebung durchgeführt werden. Dabei können die gewünschten Browser Chromium, Firefox und WebKit ausgewählt und zum Testen installiert werden. Bei der Installation wird ein Beispielttest erstellt, welcher als Vorlage für weitere Tests genutzt werden kann.

```
const EXAMPLE_ADMIN = {  
  login: 'sysop',  
  firstname: 'Admin',  
  lastname: 'User',  
  password: 'example-password',  
}
```

Abbildung 5: Beispiel für das Objekt des Administrator-Accounts

Für das Testen der meisten Funktionen der EGroupware-Anwendung wird ein bei der Installation automatisch erstellter Administrator-Account genutzt. Diesem Administrator-Account wird eine E-Mail-Adresse über IMAP hinzugefügt, damit über ihn auch die E-Mail-Funktion getestet werden kann. Um die Nutzerdaten des Accounts in den Tests zu nutzen wird der Nutzernamen und das Passwort des Administrators, wie in Abbildung 5 beispielhaft dargestellt, in der Testdatei als statisches Objekt gespeichert und kann dann in den Tests genutzt werden.

Playwright Codegen

Die Basis für fast alle Tests in dieser Studienarbeit wurde mithilfe des Playwright Tools Codegen erstellt. Dabei wird über einen Konsolenbefehl ein neuer Browser geöffnet, in dem die Interaktionen des Nutzers als Playwright-Code aufgezeichnet werden. So können große Teile des Codes für die Tests automatisch generiert werden und müssen nur noch an spezifische Anforderungen angepasst werden.

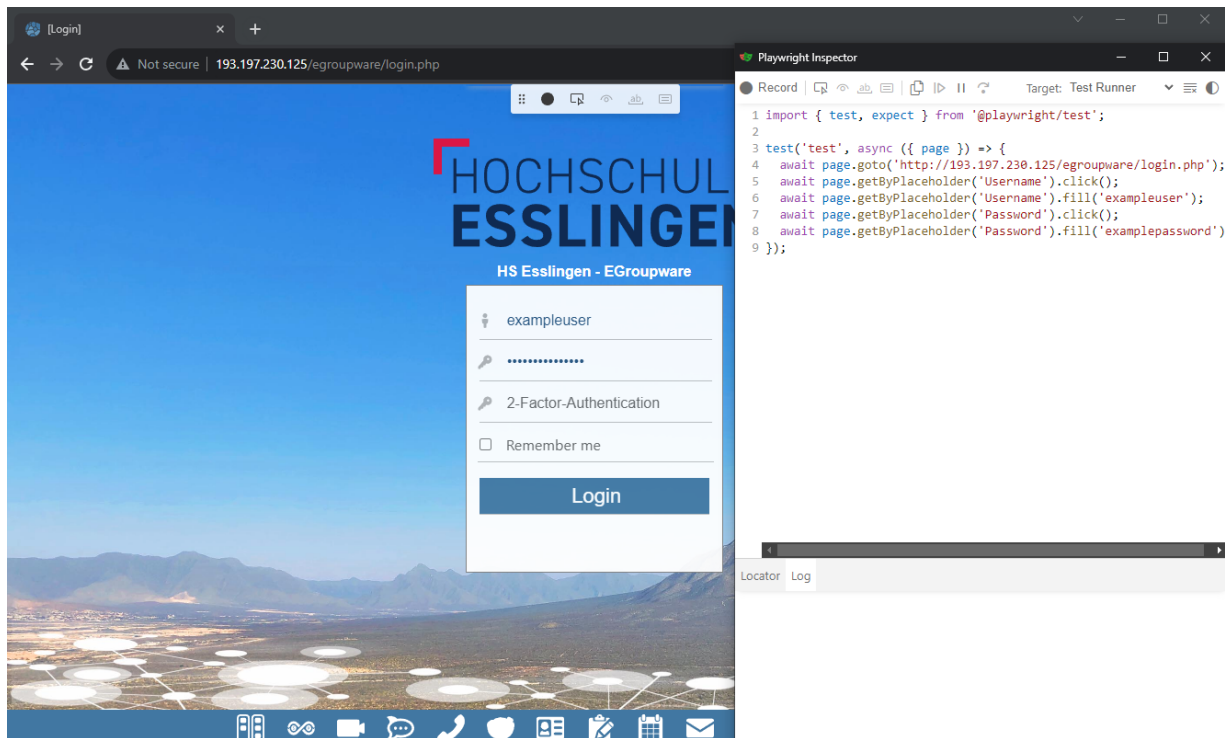


Abbildung 6: Playwright Codegen generierter Code für das Ausfüllen eines Login-Formulars

Einerseits sind Playwright und ähnliche Tools zur automatischen Codegenerierung hilfreich um schnell ohne viel Vorwissen oder Planung Tests zu erstellen, jedoch ist die Langlebigkeit und Wartbarkeit des generierten Codes fraglich, da dieser nur eine „einfache Sequenz von aufgenommenen Operationen“ (7, S. 294) ist. Playwright Codegen ist besonders gut für einfache Interaktionen, wie beispielsweise das Ausfüllen einfacher Formulare, wie in Abbildung 6 gezeigt, mit einfachen User-Interface-Elementen (UI-Elementen) wie Buttons, Textfeldern und Links geeignet. Für komplexere Interaktionen mit UI-Elementen wie beispielsweise sogenannte Datepicker, dargestellt in Abbildung 10, welche das Auswählen eines Datums über mehrere verschachtelte Interaktionen erfordern, ist die Aufnahme-Funktion des Tools weniger geeignet. In solchen Fällen müssen entweder alternative und weniger komplexe Methoden zum Auswählen des Datums gefunden werden, oder der Code muss manuell geschrieben werden. Daher mussten Teile der Tests für das Erstellen eines Termins

und das Löschen eines Nutzers manuell angepasst und teilweise komplett selbst geschrieben werden.

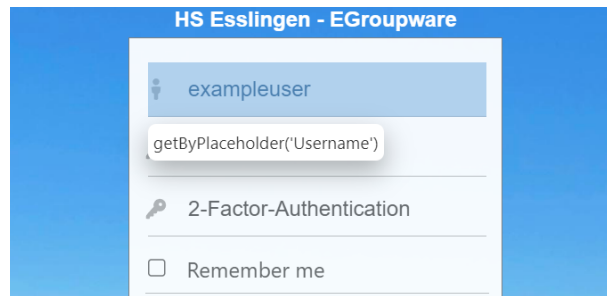


Abbildung 7: Playwright Pick-Locator Funktion zum Generieren von Selektoren für UI-Elemente

Zu der Aufnahme-Funktion des Tools gibt es noch eine weitere Funktion, die ähnlich wie Inspect-Element in Browsern funktioniert und wie in Abbildung 7 den Selektor für ein UI-Element generiert. Diese Funktion kann dann für das Erstellen der manuellen Teile der Tests genutzt werden, um schnell Selektoren für UI-Elemente zu generieren.

5.2 Implementierung der Tests

In diesem Abschnitt werden die einzelnen Tests, die in dieser Studienarbeit implementiert wurden, näher erläutert. Dabei wurden Tests zum Login, Aufrufen von E-Mails, Erstellen eines Termins und Erstellen und Löschen eines Nutzers implementiert. Alle Tests wurden in TypeScript geschrieben und können daher direkt in der Entwicklungsumgebung ausgeführt werden.

Im Folgenden werden die einzelnen Tests, deren Implementierung und Deckungsbereiche näher erläutert.

Login

```
async function login(page: Page, user) {
  await page.getByPlaceholder('Username').click();
  await page.getByPlaceholder('Username').fill(user.login);
  await page.getByPlaceholder('Password').click();
  await page.getByPlaceholder('Password').fill(user.password);
  await page.getByRole('button', { name: 'Login' }).click();
}
```

Abbildung 8: Login Test

Da alle Tests der Anwendung einen eingeloggten Nutzer benötigen, wurde zuerst ein Login-Test implementiert. Dieser erhält, wie in Abbildung 8 in der ersten Zeile zu sehen ist, ein Nutzerobjekt mit Nutzernamen und Passwort und versucht sich dann in der Anwendung einzuloggen. Der Login-Test wird zu Beginn jedes anderen Tests ausgeführt, um den Nutzer anzumelden. Dabei wird die Verfügbarkeit der Anmeldung, die Korrektheit der Login-Daten und die Existenz des Nutzers in der Anwendung getestet.

Aufrufen einer E-Mail

Bei diesem Test wird die E-Mail-Funktion der Anwendung getestet. Der Test versucht sich als der Administrator, dessen Account einen E-Mail-Account über IMAP bekommen hat, einzuloggen und ruft dann die erste E-Mail in der E-Mail-Liste auf. Dabei wird die Verbindung der Groupware zum IMAP-Server getestet.

Erstellen eines Termins

Beim Test zum Erstellen eines Termins wird sich als Administrator eingeloggt und dann ein Termin erstellt. Die Daten für diesen Termin sind ähnlich wie die Daten für den Login mit dem Administrator-Account in einem Objekt gespeichert und können so einfach in den Test eingefügt werden.

```
test('should create a simple appointment', async ({ page }) => {
  const TEST_APPOINTMENT = {
    title: 'Test Appointment',
    description: 'Test Appointment Description',
    location: 'Test Appointment Location',
    start: formatDateTime(Date.now() + 1800000), // 30 minutes from now
    duration: '1:00',
    repeat_type: 'weekly',
  };
  await login(page, SYSOP_USER);
```

Abbildung 9: Objekt zum Erstellen eines Termins innerhalb eines Tests

Jedoch wird dieses Objekt, wie in Abbildung 9 gezeigt, erst beim Ausführen des Tests erstellt, da das Datum für den Termin immer das aktuelle Datum sein soll und daher nicht statisch in einem globalen Objekt gespeichert werden kann. Dafür wird mithilfe des Zeitstempels der Funktion `Date.now()` ein Datum erstellt, welches dann mit einer Hilfsfunktion in einen String umgewandelt und in das Objekt gespeichert wird. So kann jederzeit ein Termin erstellt werden, welcher 30 Minuten nach der Ausführung des Tests stattfindet.

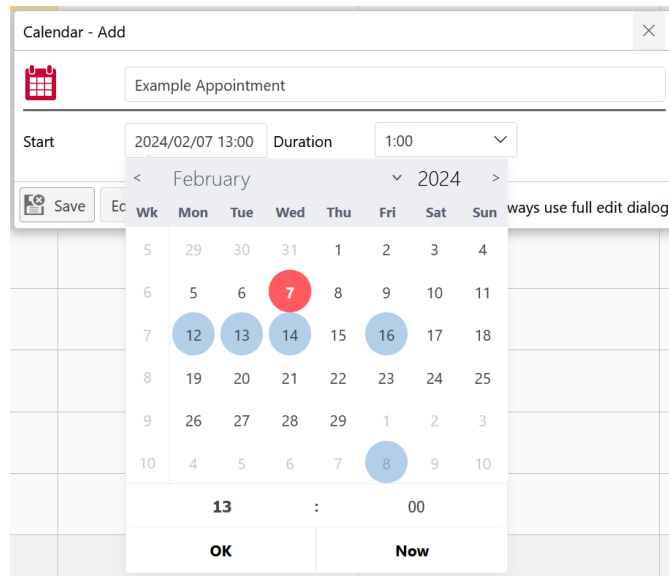


Abbildung 10: Datepicker UI-Element in EGroupware

Eine Schwierigkeit bei diesem Test war das Auswählen des Datums für den Termin. Dafür musste ein Datepicker, welcher in Abbildung 10 gezeigt ist, in der Anwendung geöffnet und das Datum ausgewählt werden. Das Datum kann zwar über einen String des Formats `YYYY/MM/DD HH:MM` in das Textfeld des Datepickers eingefügt werden, jedoch entstehen durch die Verschachtelung der UI-Elemente des Datepickers Probleme beim Auswählen des Textfelds und dem anschließenden Einfügen des Datum-Strings. Spezifisch wird das Platzhalter-Datum nicht automatisch aus dem Textfeld gelöscht, wenn dieses angeklickt wird. Daher muss als Workaround das Textfeld über Tastenbefehle geleert und dann das Datum eingefügt werden. Dafür wird, nachdem das Textfeld angeklickt wurde, mit `STRG+A` der gesamte Text im Textfeld ausgewählt und dann mit der Entfernen-Taste gelöscht. Anschließend kann der generierte Datum-String eingefügt und der Termin erstellt werden.

Erstellen und Löschen eines neuen Nutzers

Der letzte Test, der in dieser Studienarbeit implementiert wurde, ist eine Serie aus Tests, die gemeinsam die Funktionalität des Nutzermanagements abdecken sollen.

```
const EXAMPLE_USER = {  
  login: 'exampleuser',  
  firstname: 'Example',  
  lastname: 'User',  
  password: 'example-password',  
}
```

Abbildung 11: Beispiel für das Objekt zum Erstellen eines Testnutzers

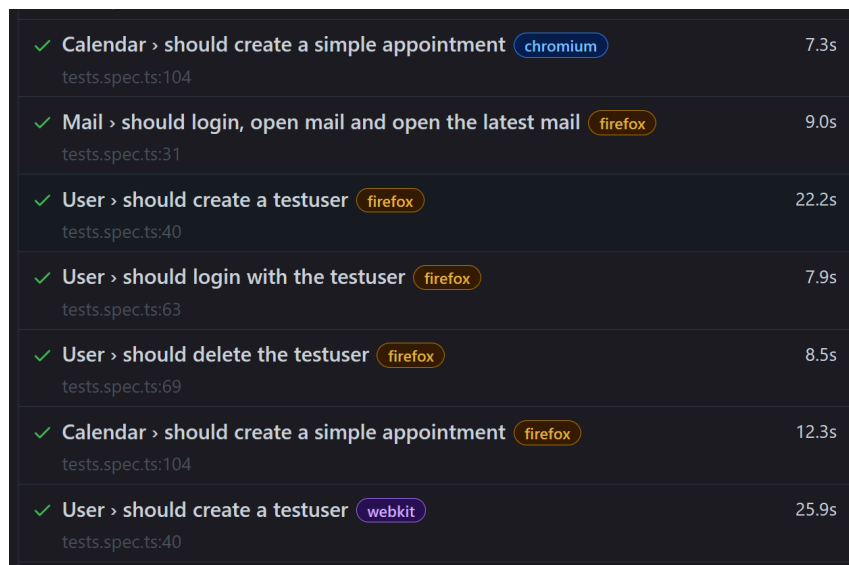
Zunächst wird mit dem Administrator-Account ein neuer Nutzer erstellt, dessen Nutzerdaten in einem statischen Objekt gespeichert sind. Ist das Erstellen des Nutzers erfolgreich, startet ein weiterer Test, der sich mit den Zugangsdaten des neuen Nutzers versucht einzuloggen. So wird sichergestellt, dass der Nutzer erfolgreich erstellt wurde und nicht nur in der lokalen Instanz des Administrators existiert. Ist das Einloggen mit dem Testnutzer erfolgreich, wird ein weiterer Test gestartet, der den Nutzer mit dem Administrator-Account wieder löscht. So kann der Test beliebig oft die gleichen Nutzerdaten zur Ausführung nutzen, ohne dass es zu Problemen mit bereits existierenden Nutzern kommt.

Bei diesem Test kam es bei der Ausführung zu einem Problem, da der Nutzer in einigen Fällen nicht sofort nach dem Erstellen in der Nutzerliste für Administratoren auftaucht. Bis der Nutzer in der Liste auftaucht, kann er sich auch nicht einloggen. Die Zeit, die der Nutzer benötigt, um in der Nutzerliste aufzutauchen ist nicht konstant und kann daher nicht durch eine einfache Wartezeit gelöst werden. Das stellt in Frage, inwiefern EGroupware für das automatisierte Erstellen von Nutzern geeignet ist und ob eine ähnliche Asynchronität auch in anderen Teilen der Anwendung auftritt.

5.3 Ausführen der Tests

Um möglichst viele Nutzerszenarien abzudecken, sollen die Tests aus den vorherigen Abschnitten in verschiedenen Browsern ausgeführt werden. Spezifisch wird jeder Test in den Browsern Chromium, Firefox und WebKit ausgeführt. Die Ausführung der Tests erfolgt dabei sequentiell, da beispielsweise das Erstellen und anschließende Löschen eines Nutzers bei paralleler Ausführung zu Problemen führen kann. Da Playwright-Tests standardmäßig parallel über mehrere Prozesse gleichzeitig ausgeführt werden, muss die maximale Anzahl der Prozesse beim Ausführen der Tests auf 1 gesetzt werden. Die Tests können dann mithilfe des folgenden Konsolenbefehls ausgeführt werden:

```
npx playwright test --workers=1
```



✓ Calendar › should create a simple appointment	chromium	7.3s
tests.spec.ts:104		
✓ Mail › should login, open mail and open the latest mail	firefox	9.0s
tests.spec.ts:31		
✓ User › should create a testuser	firefox	22.2s
tests.spec.ts:40		
✓ User › should login with the testuser	firefox	7.9s
tests.spec.ts:63		
✓ User › should delete the testuser	firefox	8.5s
tests.spec.ts:69		
✓ Calendar › should create a simple appointment	firefox	12.3s
tests.spec.ts:104		
✓ User › should create a testuser	webkit	25.9s
tests.spec.ts:40		

Abbildung 12: Testergebnisse der Playwright-Tests

Sind alle Tests einmal gelaufen, bietet Playwright eine im Browser abrufbare Übersicht über die Ergebnisse der Tests. Die Übersicht zeigt, wie in Abbildung 12 dargestellt, für jeden Test, ob dieser erfolgreich war, wie lange er gebraucht hat und in welchem Browser er ausgeführt wurde. Schlagen Tests fehl, wird in der Übersicht zu jedem fehlgeschlagenen Test eine Fehlermeldung angezeigt, die weitere Informationen bietet, wo genau im Test der Fehler aufgetreten ist.

Bei der Ausführung der Tests auf verschiedenen Browsern konnten keine konstanten spezifischen Probleme auf einem der Browser festgestellt werden. Daher ist davon auszugehen, dass die Anwendung in allen gängigen Browsern gleich gut funktioniert.

6 Zusammenfassung

In diesem Kapitel wird ein Fazit über den Verlauf des Studienprojekts und das im Rahmen des Studienprojekts untersuchte Groupware-System als Lösung für die Hochschule Esslingen gezogen. Zuletzt wird ein Ausblick auf mögliche nächste Schritte in der Suche nach einer Groupware-Lösung für die Hochschule Esslingen gegeben.

6.1 Fazit

Das Ziel des Studienprojekts war es, ein Groupware-System zu finden und zu testen, das die Anforderungen der Hochschule Esslingen erfüllt. Dabei wurden die Anforderungen an das Groupware-System in einem ersten Schritt definiert und anschließend die verfügbaren Groupware-Systeme recherchiert. Nachdem eins der Groupware-Systeme ausgewählt wurde, wurde es auf einer Cloud-Instanz installiert und konfiguriert. Zuletzt wurden automatisierte End-to-End-Tests für das Groupware-System implementiert und ausgeführt.

Mit der Installation und dem Testen eines ersten Groupware-Systems wurde das Ziel des Studienprojekts erreicht. Das untersuchte Groupware-System, EGroupware, erfüllt dabei die funktionalen Anforderungen der Hochschule an ein Groupware-System, wie E-Mails, Kalender mit Terminplanung und Kontakte. Auch die nicht-funktionale Anforderung der Eigenadministration wurde erfüllt, da die Installation und Konfiguration des Systems auf einer Cloud-Instanz durchgeführt werden konnte.

Ein Faktor, bei dem bei EGroupware noch Zweifel aufwirft, ist die Möglichkeit zur Automatisierung von Teilen des Systems. Da bei Tests mit Playwright zum Erstellen und Löschen von Nutzern Asynchronitäten in den angezeigten Nutzer-Accounts auftraten, ist in Frage zu stellen, ob das System für ausführliche Automatisierung geeignet ist. Daher kann ohne weitere ausführliche Tests noch nicht eindeutig gesagt werden, ob EGroupware die Anforderungen der Hochschule Esslingen aus den recherchierten Groupware-Systemen am besten erfüllt.

6.2 Ausblick

In diesem Studienprojekt wurde die Basis der Suche nach einem neuen Groupware-System für die Hochschule Esslingen gelegt. Dabei wurden die Anforderungen an das Groupware-System definiert und ein erstes Groupware-System installiert und getestet.

Im nächsten Schritt soll, möglicherweise in weiteren Studienprojekten, die Recherche nach einem Groupware-System mit der Installation und dem Testen weiterer Systeme fortgesetzt werden. Alle der in diesem Studienprojekt betrachteten Groupware-Systeme erfüllen die grundsätzlichen Anforderungen der Hochschule an ein neues Groupware-System. Daher könnte es sinnvoll sein, in Zukunft noch Kolab, Horde und Sogo, die bereits von deutlich mehr Hochschulen genutzt werden als EGroupware, als mögliche Lösungen für die Hochschule Esslingen zu betrachten. Aufgrund der Recherche dieser Studienarbeit würde sich als nächstes zu untersuchendes System besonders Sogo anbieten, da es von den untersuchten Systemen von den meisten Hochschulen als Groupware-System genutzt wird.

Falls EGroupware weiterhin als Lösung für die Hochschule Esslingen in Betracht gezogen wird, sollte die mögliche Automatisierung von Teilen des Systems genauer untersucht werden. Beispielsweise sollte die Robustheit des Systems beim Erstellen vieler Nutzer auf einmal getestet werden, um sicherzustellen, dass das System die Anforderungen der Hochschule Esslingen erfüllt. Dabei sollte EGroupware auch auf leistungstärkeren Systemen als der in diesem Studienprojekt verwendeten Cloud-Instanz getestet werden, um sicherzustellen, dass auftretende Asynchronitäten nicht durch Leistungsmangel verursacht wurden.

Literatur

1. MICROSOFT. *Playwright Homepage* [online]. Microsoft, [o. D.] [besucht am 2024-01-29]. Abger. unter: <https://playwright.dev/>.
2. SCHMITT, Jacob. *What is end-to-end testing?* [online]. 2023. [besucht am 2024-02-08]. Abger. unter: <https://circleci.com/blog/what-is-end-to-end-testing/>. 07.04.2024.
3. KOLAB. *Kolab Homepage* [online]. [o. D.]. [besucht am 2024-01-29]. Abger. unter: <https://kolab.org/>.
4. HORDE. *Horde Homepage* [online]. [o. D.]. [besucht am 2024-01-29]. Abger. unter: <https://www.horde.org/apps/groupware>.
5. SOGO. *Sogo Homepage* [online]. [o. D.]. [besucht am 2024-01-29]. Abger. unter: <https://www.sogo.nu/>.
6. EGROUWARE. *EGroupware Installation* [online]. EGroupware, [o. D.] [besucht am 2024-01-29]. Abger. unter: <https://github.com/EGroupware/egroupware/wiki/Installation-using-egroupware-docker-RPM-DEB-package>.
7. KIRINUKI, Hiroyuki; TANNO, Haruto. Automating End-to-End Web Testing via Manual Testing. *Journal of Information Processing*. 2022, Jg. 30, S. 294–306. Abger. unter DOI: 10.2197/ipsjjip.30.294.