

COSPAR Communication System

Prüfling: Abdullah Diab
Geburtsdatum: 01.01.2006
Ausbildungsberuf: Fachinformatiker für Anwendungsentwicklung
Ausbildungsbetrieb: AFZ - Aus- und Fortbildungszentrum Bremen
Ausbildungsstelle: ZARM - Zentrum für angewandte Raumfahrttechnologie und Mikrogravitation
Prüflings-Nr.: 40016

Eidesstattliche Erklärung

Hiermit versichere ich, dass das Projekt mit dem Titel „COSPAR Communication System“ und die dazugehörige Dokumentation in Einzelarbeit von meiner Person angefertigt wurde. Diese Dokumentation wurde in der Vergangenheit nicht bei Prüfungen zur Bewertung/Begutachtung vorgelegt.

Bremen, den _____

Ort, Datum und Unterschrift des Prüfungsteilnehmers

Inhaltsverzeichnis

1	Einleitung	1
1.1	Der Ausbildungsbetrieb	1
1.2	Zielsetzung des Projekts	1
1.3	Was ist COSPAR?	1
1.4	Die Rolle der MSOs, DOs und Autoren	2
2	Anforderungsanalyse	3
2.1	Ist-Analyse	3
2.1.1	Aktuelle Kommunikationsweise der MSOs/DOs	3
2.1.2	Probleme und Schwachstellen	3
2.2	Soll-Analyse	3
2.2.1	Funktionale Anforderungen	3
2.2.2	Nicht-funktionale Anforderungen	3
2.3	Wirtschaftlichkeitsbetrachtung	4
3	Entwurfsphase	5
3.1	Systemarchitektur und Datenfluss	5
3.2	Technische Architektur	5
3.2.1	Analyse der bestehenden Codebasis	5
3.2.2	Wahl der Technologien	5
3.2.3	Anbindung an die bestehende Datenbank	6
3.3	Konzeption der Features	6
3.3.1	Autorenfilterung und -auswahl	6
3.3.2	Externe E-Mail-Client-Integration	6
3.3.3	Benutzeroberflächen-Design	7
3.3.4	Integration in bestehende COSPAR-Infrastruktur	7
3.4	UI/UX Design	7
3.4.1	Wireframes und Mockups	7
3.4.2	Responsives Designkonzept	7
3.5	Projektplanung	8
3.5.1	Zeitplan	8
3.5.2	Meilensteine	8
4	Projektdurchführung	9
4.1	Entwicklungsumgebung einrichten	9
4.2	Backend-Implementierung	9
4.2.1	Datenbankzugriffe	9
4.2.2	MSO/DO-Authentifizierung	9

4.2.3	E-Mail-System-Integration	10
4.3	Frontend-Implementierung	10
4.3.1	Filter-Wizard-Interface	10
4.3.2	Autorenkarten-Interface	10
4.3.3	Kommunikationsübersicht-Interface	10
4.3.4	E-Mail-Client-Integration-Interface	11
4.4	Styling und Design	11
4.4.1	CSS-Implementierung	11
4.4.2	JavaScript-Funktionalitäten	11
4.5	Testing	12
4.5.1	Funktionalitätstests	12
4.5.2	Usability Tests	12
5	Projektabschluss	13
5.1	Soll-Ist-Vergleich	13
5.2	Ausblick und Nutzen für COSPAR	13
5.3	Reflexion	14
5.4	Fazit	14
A	Quellcode-Auszüge	15
A.1	PHP Backend - UserHelper.php	15
A.2	JavaScript Frontend - Filter Wizard	17
A.3	E-Mail-Client-Integration	23
B	Screenshots	24
C	Testprotokolle	25
C.1	Funktionalitätstests - Ergebnisse	25
C.2	Usability-Test-Ergebnisse	25
D	Projektplan (Gantt-Diagramm)	25
E	Glossar	26

Abbildungsverzeichnis

1 Einleitung

1.1 Der Ausbildungsbetrieb

Das Zentrum für angewandte Raumfahrttechnologie und Mikrogravitation (ZARM) der Universität Bremen ist eine international anerkannte Forschungseinrichtung, die sich auf Grundlagenforschung und angewandte Forschung in der Raumfahrttechnologie und Mikrogravitationsforschung spezialisiert hat. Als interdisziplinäres Zentrum verbindet das ZARM Expertise aus den Bereichen Physik, Ingenieurwissenschaften und Materialwissenschaften.

Das ZARM spielt eine bedeutende Rolle in der internationalen Raumfahrtforschung und unterhält enge Kooperationen mit führenden Raumfahrtagenturen wie der ESA, NASA und DLR. Durch diese Vernetzung ist das ZARM auch aktiv in wissenschaftlichen Organisationen wie COSPAR (Committee on Space Research) vertreten, wo Mitarbeiter wichtige Funktionen als Main Scientific Organizers (MSOs) und Deputy Organizers (DOs) übernehmen.

Im Rahmen der Ausbildung werden praktische IT-Projekte durchgeführt, die reale Problemstellungen aus dem wissenschaftlichen Betrieb aufgreifen und digitale Lösungen entwickeln. Das COSPAR Communication System Projekt ist ein solches Ausbildungsprojekt, das eine konkrete Herausforderung im COSPAR-Organisationsprozess adressiert.

1.2 Zielsetzung des Projekts

Das COSPAR Communication System Projekt verfolgt mehrere zentrale Ziele zur Verbesserung der Kommunikationsprozesse zwischen MSOs/DOs und ihren Session-Autoren. Ein Hauptziel ist die deutliche Steigerung der Effizienz, sodass MSOs und DOs ihre Autoren wesentlich schneller kontaktieren können als bisher. Dies wird durch eine Bulk-E-Mail-Funktionalität erreicht, die es ermöglicht, mehrere Autoren gleichzeitig zu kontaktieren.

Ein weiterer wichtiger Aspekt ist die Entwicklung intelligenter Filteroptionen, die eine präzise Auswahl nach verschiedenen Kriterien ermöglichen. Diese umfassen die Filterung nach Sessions, Autorentypen wie Presenting Authors und Co-Authors sowie nach dem Präsentationsstatus. Dadurch können MSOs und DOs gezielt bestimmte Autorengruppen ansprechen.

Das Projekt eliminiert zeitaufwändige manuelle Prozesse durch automatisierte Systeme, die direkt auf die COSPAR-Datenbank zugreifen. Die nahtlose Integration in die bestehende COSPAR-Webinfrastruktur gewährleistet, dass das neue System harmonisch mit den vorhandenen Arbeitsabläufen funktioniert, ohne zusätzliche Komplexität für die Benutzer zu verursachen.

1.3 Was ist COSPAR?

COSPAR (Committee on Space Research) ist eine internationale wissenschaftliche Organisation, die 1958 vom International Council for Science (ICSU) gegründet wurde und als weltweites Forum für den wissenschaftlichen Austausch in der Weltraumforschung dient. Die Organisation koordiniert internationale Bemühungen in allen Bereichen der Raumfahrtwissenschaften und fördert die Zusammenarbeit

zwischen verschiedenen Raumfahrtagenturen und Forschungseinrichtungen weltweit. Das wichtigste Ereignis von COSPAR sind die alle zwei Jahre stattfindenden Scientific Assemblies (Wissenschaftliche Versammlungen), bei denen Tausende von Wissenschaftlern aus aller Welt zusammenkommen, um aktuelle Forschungsergebnisse zu präsentieren und zu diskutieren. Diese Konferenzen sind nach wissenschaftlichen Kommissionen und Panels strukturiert, die verschiedene Bereiche der Weltraumforschung abdecken - von Planetenforschung über Astrophysik bis hin zu Weltraumtechnologie. COSPAR entwickelt wichtige Standards für Planetenschutz, koordiniert Datenformat-Richtlinien und organisiert den strukturierten Prozess der wissenschaftlichen Beitragsverwaltung über webbasierte Systeme. Die komplexe Organisation dieser internationalen Veranstaltungen erfordert ein effizientes Management-System, in dem die Main Scientific Organizers (MSOs) und Deputy Organizers (DOs) eine zentrale Rolle spielen, da sie die Schnittstelle zwischen der COSPAR-Führung und der wissenschaftlichen Gemeinschaft bilden.

1.4 Die Rolle der MSOs, DOs und Autoren

Main Scientific Organizers (MSOs): Hauptverantwortliche für wissenschaftliche Sessions, die den gesamten Ablauf von der Abstract-Bewertung bis zur Session-Durchführung koordinieren. Sie müssen regelmäßig mit ihren Autoren kommunizieren, um Updates zu Präsentationen, Zeitplänen und technischen Anforderungen zu übermitteln.

Deputy Organizers (DOs): Unterstützen MSOs bei der Session-Organisation und fungieren als zusätzliche Ansprechpartner. Sie benötigen ebenfalls direkten Zugang zu Autorenlisten für koordinierte Kommunikation.

Autoren: Wissenschaftler, die Abstracts einreichen und in verschiedenen Kategorien auftreten:

- **Presenting Authors:** Hauptautoren, die ihre Forschung präsentieren
- **Co-Authors:** Mitautoren ohne Präsentationspflicht

Kommunikationsbeziehungen: MSOs/DOs müssen Autoren über Präsentationsdetails, Upload-Deadlines, technische Anforderungen und Zeitplanänderungen informieren. Bisher erfolgte dies manuell und zeitaufwändig.

2 Anforderungsanalyse

2.1 Ist-Analyse

2.1.1 Aktuelle Kommunikationsweise der MSOs/DOs

Der bisherige Kommunikationsprozess erfolgt vollständig manuell und ist sehr zeitaufwändig. MSOs und DOs müssen zunächst ihr E-Mail-Programm wie Outlook oder Thunderbird öffnen und dann alle benötigten Autoren-E-Mail-Adressen einzeln aus der COSPAR-Datenbank kopieren. Für jede Kommunikation muss eine separate E-Mail erstellt werden, da kein systematisches Filterungssystem für verschiedene Autorengruppen existiert. Dieser Prozess wird besonders bei Sessions mit vielen Teilnehmern sehr zeitaufwändig und fehleranfällig.

2.1.2 Probleme und Schwachstellen

Die manuelle Vorgehensweise führt zu verschiedenen Problemen. Tippfehler bei der E-Mail-Adressen-Eingabe können dazu führen, dass wichtige Nachrichten nicht zugestellt werden. Bei umfangreichen Autorenlisten werden häufig einzelne Autoren vergessen, was zu unvollständiger Kommunikation führt. Die repetitiven Aufgaben bei jeder Kommunikationsrunde verschwenden wertvolle Arbeitszeit. Zusätzlich fehlt jede Art von Dokumentation oder Historienführung über gesendete E-Mails, und es gibt keine Möglichkeit zur gezielten Filterung nach Autorentypen oder Session-Eigenschaften.

2.2 Soll-Analyse

2.2.1 Funktionale Anforderungen

Das neue System soll eine sichere MSO/DO-Authentifizierung mit Berechtigungsprüfung für spezifische Sessions bieten. Kern des Systems sind session-basierte Autorenlisten, die automatisch alle Autoren einer bestimmten Session anzeigen. Erweiterte Filteroptionen ermöglichen die Auswahl nach Autorentyp (Presenting Authors oder Co-Authors), Präsentationstyp (Oral oder Poster Präsentations) und Upload-Status der Präsentationen. Die Integration mit externen E-Mail-Clients erfolgt über automatische Mailto-Link-Generierung, während Such- und Filterfunktionen eine Echtzeit-Suche in Autorenlisten ermöglichen. Das responsive Design gewährleistet die Funktionalität auf verschiedenen Geräten.

2.2.2 Nicht-funktionale Anforderungen

Das System muss auch bei hunderten von Autoren pro Session eine schnelle Filterung gewährleisten und durch intuitive Bedienung maximal 3 Klicks zur E-Mail-Erstellung benötigen. Sichere Authentifizierung und datenschutzkonforme Handhabung von E-Mail-Adressen sind essentiell. Die browser-übergreifende Kompatibilität muss für Chrome, Firefox, Safari und Edge sichergestellt werden, während die nahtlose Integration in die bestehende COSPAR-Infrastruktur ohne Störung anderer

Systeme erfolgen soll. Das System muss außerdem mit wachsenden Teilnehmerzahlen skalieren können.

2.3 Wirtschaftlichkeitsbetrachtung

Die Wirtschaftlichkeitsanalyse zeigt einen deutlichen Nutzen gegenüber dem Entwicklungsaufwand. Die größte Zeitersparnis liegt in der Reduktion der Kommunikationszeit von über 30 Minuten auf unter 2 Minuten pro Session. Gleichzeitig werden manuelle Eingabefehler bei E-Mail-Adressen eliminiert, was die Zuverlässigkeit der Kommunikation erheblich verbessert. Der Entwicklungsaufwand von etwa 35 Stunden führt zu langfristigen Effizienzgewinnen, wobei sich die Investition bereits nach wenigen COSPAR-Assemblies durch die eingesparte Arbeitszeit amortisiert.

3 Entwurfsphase

3.1 Systemarchitektur und Datenfluss

Das COSPAR Communication System basiert auf einer modularen Architektur, die sich nahtlos in die bestehende COSPAR-Infrastruktur integriert:

Datenfluss:

1. MSO/DO-Anmeldung über bestehendes COSPAR-Login-System
2. Session-Identifikation und Berechtigungsprüfung
3. Datenbankabfrage für Session-spezifische Autoren
4. Anwendung von Filterkriterien (AJAX-basiert)
5. Generierung von Mailto-Links für externe E-Mail-Clients
6. Integration mit bestehenden E-Mail-Programmen der Benutzer

3.2 Technische Architektur

3.2.1 Analyse der bestehenden Codebasis

Die vorhandene COSPAR-Webseite verwendet folgende Technologien:

- **Backend:** PHP mit MySQL-Datenbank
- **Frontend:** HTML, CSS, JavaScript
- **Framework:** Eigenes MVC-ähnliches System
- **Authentifizierung:** Session-basierte Benutzeranmeldung
- **Datenbankzugriff:** mysqli-Verbindungen

3.2.2 Wahl der Technologien

Die Technologiewahl orientiert sich an der bestehenden COSPAR-Infrastruktur:

- **Frontend:** HTML5, CSS3, Vanilla JavaScript für maximale Kompatibilität
- **Backend:** PHP 7.4+ für Konsistenz mit bestehenden Systemen
- **Datenbank:** MySQL mit bestehenden COSPAR-Tabellen
- **AJAX:** Für dynamische Filterung ohne Seitenneuladung
- **Responsive Design:** CSS Grid und Flexbox für Gerätekompatibilität

3.2.3 Anbindung an die bestehende Datenbank

Integration erfolgt über bestehende Tabellen:

- **user:** Benutzerinformationen und Authentifizierung
- **sessions:** Session-Details und MSO/DO-Zuordnungen
- **abstracts:** Abstract-Informationen und Autorenverknüpfungen
- **authors:** Autoreninformationen und E-Mail-Adressen
- **presentations:** Upload-Status und Präsentationstypen

3.3 Konzeption der Features

3.3.1 Autorenfilterung und -auswahl

Konzept der intelligenten Autorenfilterung:

- **Session-basierte Filterung:** Automatische Anzeige nur relevanter Autoren (A0.1, A0.2, etc.)
- **Autorentyp-Filter:** Unterscheidung zwischen Presenting Authors und Co-Authors
- **Kombinierte Filterkriterien:** Mehrere Filter gleichzeitig anwendbar
- **Ergebnistabelle:** Übersichtliche Darstellung mit Autoreninformationen
- **Auswahl-Checkboxes:** Individuelle Empfängersteuerung möglich

3.3.2 Externe E-Mail-Client-Integration

Konzept der nahtlosen Integration mit bestehenden E-Mail-Programmen:

- **Mailto-Link-Generierung:** Automatische Erstellung von Links für externe E-Mail-Programme
- **Empfängerübernahme:** Automatische Übernahme gefilterter E-Mail-Adressen
- **Client-Kompatibilität:** Funktioniert mit Outlook, Thunderbird, Apple Mail, etc.
- **Bulk-E-Mail-Support:** Mehrere Empfänger in BCC für Datenschutz
- **Benutzerfreundlichkeit:** Ein-Klick-Lösung für E-Mail-Erstellung

3.3.3 Benutzeroberflächen-Design

Moderne und intuitive Benutzeroberfläche:

- **Session-Header:** Farbkodierte Darstellung der Session-Informationen
- **Autorenkarten:** Card-basierte Anzeige mit wichtigen Informationen
- **Echtzeit-Suche:** Sofortige Filterung bei Texteingabe
- **Filter-Wizard:** Mehrstufiger Assistent für komplexe Filterkriterien
- **Responsive Design:** Optimale Darstellung auf allen Geräten

3.3.4 Integration in bestehende COSPAR-Infrastruktur

Nahtlose Einbindung in das bestehende System:

- **Embedded-Modus:** Integration als Komponente in bestehende Seiten
- **Session-Management:** Nutzung der vorhandenen Authentifizierung
- **Datenkonsistenz:** Direkter Zugriff auf COSPAR-Datenbank
- **Design-Konsistenz:** Anpassung an bestehende Optik und Bedienung

3.4 UI/UX Design

3.4.1 Wireframes und Mockups

Das Design orientiert sich an modernen Web-Standards:

- **Card-basierte Layouts:** Übersichtliche Darstellung von Autoreninformationen
- **Modal-Dialogs:** Filter-Wizard als überlagertes Fenster
- **Farbkodierung:** Session-spezifische Farben für bessere Orientierung
- **Responsive Grid:** Automatische Anpassung an verschiedene Bildschirmgrößen

3.4.2 Responsives Designkonzept

- **Mobile-First-Ansatz:** Optimierung für kleinste Bildschirme zuerst
- **Breakpoints:** 768px (Tablet), 1024px (Desktop), 1200px (Large Desktop)
- **Touch-Optimierung:** Größere Buttons und Touch-Targets für mobile Geräte
- **Progressive Enhancement:** Grundfunktionalität ohne JavaScript verfügbar

3.5 Projektplanung

3.5.1 Zeitplan

Phase	Dauer	Meilensteine
Analyse und Planung	1 Woche	Anforderungen definiert
Design und Architektur	1 Woche	UI-Mockups und Datenbankschema
Backend-Entwicklung	2 Wochen	Datenbankanbindung und Filterlogik
Frontend-Entwicklung	2 Wochen	UI-Komponenten und AJAX-Integration
Testing und Optimierung	1 Woche	Funktions- und Usability-Tests
Gesamt	7 Wochen	System produktionsreif

3.5.2 Meilensteine

- **M1:** Analyse abgeschlossen - Anforderungen vollständig dokumentiert
- **M2:** Design fertig - UI-Mockups und Datenbankschema erstellt
- **M3:** Backend-Grundfunktionen implementiert - Datenbankanbindung und Filterlogik funktionsfähig
- **M4:** Frontend-Integration abgeschlossen - Vollständige Benutzeroberfläche verfügbar
- **M5:** Testing und Deployment - System getestet und produktionsreif

4 Projektdurchführung

4.1 Entwicklungsumgebung einrichten

Für die Entwicklung des CosparMail Systems wurde folgende Umgebung eingerichtet:

- **Lokale Entwicklung:** XAMPP für Windows mit Apache, MySQL und PHP 7.4+
- **Code-Editor:** Visual Studio Code mit PHP- und JavaScript-Extensions
- **Versionskontrolle:** Git für Versionierung und Backup
- **Browser-Entwicklertools:** Chrome DevTools für Frontend-Debugging und Testing
- **Datenbankzugriff:** phpMyAdmin für Datenbankmanagement und SQL-Entwicklung

4.2 Backend-Implementierung

4.2.1 Datenbankzugriffe

Die Datenbankanbindung erfolgt über die bestehende COSPAR-Infrastruktur:

- **Sichere Verbindungen:** Nutzung der vorhandenen mysqli-Verbindungen mit Prepared Statements
- **Autorenfilterung:** Komplexe SQL-Abfragen zur Verknüpfung von Sessions, Abstracts und Autoren
- **Integration:** Anbindung an bestehende Tabellen (user, sessions, abstracts, authors)
- **Performance-Optimierung:** Indexierte Abfragen und effiziente JOIN-Operationen

4.2.2 MSO/DO-Authentifizierung

- **Integration:** Nutzung des bestehenden COSPAR-Login-Systems
- **Session-Management:** Sichere PHP-Sessions mit Timeout-Funktionalität
- **Berechtigungsprüfung:** Überprüfung der MSO/DO-Rollen für spezifische Sessions
- **Konstanten:** Definition von FUNCTION_ID_MSO (5) und FUNCTION_ID_DO (4) für Rollenprüfung

4.2.3 E-Mail-System-Integration

- **Mailto-Link-Generierung:** Automatische Erstellung von Links für externe E-Mail-Clients
- **Empfängerverwaltung:** Sichere Handhabung von E-Mail-Adressen mit Datenschutz-Compliance
- **Bulk-E-Mail-Support:** BCC-Funktionalität für Massenversand
- **Client-Kompatibilität:** Testing mit verschiedenen E-Mail-Programmen

4.3 Frontend-Implementierung

4.3.1 Filter-Wizard-Interface

Der mehrstufige Filter-Assistent bietet eine benutzerfreundliche Oberfläche:

- **Schritt 1:** Autorentyp-Auswahl (All Authors, Presenting Authors, Co-Authors)
- **Schritt 2:** Präsentationstyp-Filter (All, Oral Presentations, Poster Presentations)
- **Schritt 3:** Upload-Status-Filter (All, With Presentation, Without Presentation)
- **Schritt 4:** Zusammenfassung und Bestätigung der Filterkriterien
- **Features:** Modal-Dialog mit Fortschrittsanzeige, Navigation zwischen Schritten, Zurücksetzen-Funktion

4.3.2 Autorenkarten-Interface

Übersichtliche Darstellung der gefilterten Autoren:

- **Card-basiertes Layout:** Jeder Autor wird in einer eigenen Karte dargestellt
- **Wichtige Informationen:** Name, E-Mail, Abstract-Titel, Autorentyp
- **Visual Feedback:** Farbkodierung je nach Session und Autorentyp
- **Interaktive Elemente:** Hover-Effekte und Auswahlmöglichkeiten

4.3.3 Kommunikationsübersicht-Interface

- **Session-Header:** Farbkodierte Anzeige der Session-Informationen
- **Suchfunktion:** Echtzeit-Filterung während der Texteingabe
- **Filter-Controls:** Zentrale Steuerung aller Filteroptionen
- **Responsive Layout:** Automatische Anpassung an verschiedene Bildschirmgrößen

4.3.4 E-Mail-Client-Integration-Interface

- **Action-Buttons:** Zentrale Schaltfläche für E-Mail-Client-Öffnung
- **Empfänger-Vorschau:** Anzeige der Anzahl ausgewählter Autoren
- **Status-Feedback:** Bestätigungen und Fehlermeldungen
- **Accessibility:** Keyboard-Navigation und Screen-Reader-Unterstützung
- **Externe Integration:** Öffnung des Standard-E-Mail-Clients des Benutzers
- **Vorgefüllte Daten:** Automatische Übernahme von Empfängern und Basisinformationen
- **Flexibilität:** Vollständige Bearbeitung im gewohnten E-Mail-Programm möglich
- **Datenschutz:** BCC-Verwendung zum Schutz der Empfänger-E-Mail-Adressen

4.4 Styling und Design

4.4.1 CSS-Implementierung

- **Responsive CSS-Design:** Mobile-First-Ansatz mit flexiblen Grid-Layouts
- **Integration:** Nahtlose Einbindung in das bestehende COSPAR-Design
- **Cross-Browser-Kompatibilität:** Testing und Optimierung für alle gängigen Browser
- **Modulares CSS:** Separate Stylesheets für verschiedene Komponenten
- **Performance:** Optimierte CSS-Dateien mit minimaler Dateigröße

4.4.2 JavaScript-Funktionalitäten

- **Event-Handling:** Interaktive Filter- und Suchfunktionen
- **AJAX-Aufrufe:** Asynchrone Datenübertragung für dynamische Inhalte ohne Seitenneuladung
- **Form-Validierung:** Client-seitige Validierung für bessere Benutzerfreundlichkeit
- **Modal-Management:** Verwaltung des Filter-Wizard-Dialogs
- **Responsive Utilities:** JavaScript-basierte Anpassungen für verschiedene Geräte

4.5 Testing

4.5.1 Funktionalitätstests

- **Filter-Funktionalität:** Überprüfung aller Filterkombinationen mit verschiedenen Datensets
- **E-Mail-Integration:** Tests der Mailto-Link-Generierung mit verschiedenen E-Mail-Clients
- **Cross-Browser-Tests:** Kompatibilitätstests in Chrome, Firefox, Safari und Edge
- **Performance-Tests:** Lasttest mit großen Autorenlisten (500+ Autoren)
- **AJAX-Funktionalität:** Tests der asynchronen Datenübertragung und Fehlerbehandlung

4.5.2 Usability Tests

- **MSO/DO-Feedback:** Direktes Testing mit echten Benutzern aus dem ZARM
- **UI/UX-Verbesserungen:** Iterative Anpassungen basierend auf Benutzerfeedback
- **Accessibility:** Tests der Barrierefreiheit und Keyboard-Navigation
- **Workflow-Tests:** Überprüfung des gesamten Kommunikationsprozesses
- **Performance-Wahrnehmung:** Subjektive Bewertung der Systemgeschwindigkeit

5 Projektabschluss

5.1 Soll-Ist-Vergleich

Anforderung	Status	Erfüllung
MSO/DO-Authentifizierung	Vollständig	100%
Session-basierte Autorenlisten	Vollständig	100%
Autorentyp-Filterung	Vollständig	100%
Präsentationstyp-Filter	Vollständig	100%
Upload-Status-Filter	Vollständig	100%
E-Mail-Client-Integration	Vollständig	100%
Responsive Design	Vollständig	100%
Performance-Optimierung	Vollständig	100%

Zusätzlich implementierte Features:

- Echtzeit-Autorensuche mit sofortiger Filterung
- Enhanced Filter-Wizard mit mehrstufiger Navigation
- Improved User Experience mit Loading-States und Feedback-Mechanismen
- Erweiterte Responsive-Design-Features für optimale mobile Nutzung

5.2 Ausblick und Nutzen für COSPAR

Das Communication System bringt erhebliche Verbesserungen für COSPAR:

Direkte Vorteile:

- **Zeitersparnis:** Reduktion der Kommunikationszeit von 30+ Minuten auf unter 2 Minuten
- **Fehlerreduktion:** Eliminierung manueller Eingabefehler bei E-Mail-Adressen
- **Skalierbarkeit:** System kann mit wachsenden Teilnehmerzahlen problemlos umgehen
- **Benutzerakzeptanz:** Positive Resonanz von MSOs/DOs auf das neue System

Erweiterungsmöglichkeiten:

- **Kurzfristig:** Integration von E-Mail-Templates für häufige Nachrichten
- **Mittelfristig:** Implementierung eines internen E-Mail-Systems
- **Langfristig:** Erweiterte Reporting-Funktionen und Mobile App-Entwicklung

5.3 Reflexion

Technische Erkenntnisse:

- Vertiefung der PHP-Kenntnisse, besonders in Datenbankintegration und AJAX-Handling
- JavaScript-Expertise in DOM-Manipulation und asynchroner Programmierung
- MySQL-Optimierung und komplexe Query-Entwicklung
- CSS-Frameworks und responsive Design-Prinzipien
- Bedeutung der Browser-Kompatibilität und Cross-Platform-Testing

Projektmanagement-Erfahrungen:

- Agile Entwicklungsmethoden und iterative Verbesserung
- Stakeholder-Kommunikation und Anforderungsmanagement
- Testing-Strategien und Qualitätssicherung
- Zeitmanagement und Meilenstein-Planung

Kommunikation mit Stakeholdern:

- Regelmäßige Abstimmung mit MSOs/DOs für Anforderungvalidierung
- Benutzer-zentrierte Entwicklung basierend auf realem Feedback
- Dokumentation für zukünftige Entwickler und Administratoren
- Präsentation technischer Konzepte für nicht-technische Stakeholder

5.4 Fazit

Das COSPAR Communication System Projekt wurde erfolgreich abgeschlossen und erfüllt alle gesteckten Ziele. Die Lösung verbessert die Kommunikationseffizienz zwischen MSOs/DOs und ihren Autoren erheblich und trägt zur Optimierung der COSPAR-Organisationsprozesse bei.

Zielerreichung: Alle funktionalen und nicht-funktionalen Anforderungen wurden vollständig implementiert und getestet. Das System ist produktionsreif und wird bereits aktiv von COSPAR-Mitarbeitern genutzt.

Persönliche Bewertung: Das Projekt bot wertvolle Erfahrungen in der Entwicklung praxisrelevanter Webapplikationen und verdeutlichte die Bedeutung benutzerorientierter Softwareentwicklung. Die Herausforderung, ein System in eine bestehende Infrastruktur zu integrieren, erweiterte das Verständnis für Enterprise-Entwicklung erheblich.

Danksagungen: Besonderen Dank an die MSOs/DOs des ZARM für ihre aktive Teilnahme am Testing-Prozess und ihre konstruktiven Rückmeldungen, sowie an die Betreuer für die fachliche Unterstützung während der Projektlaufzeit.

A Quellcode-Auszüge

A.1 PHP Backend - UserHelper.php

```
1 <?php
2 /**
3  * User Helper Functions - Session-spezifische Filterung
4  */
5
6 // Permission system constants
7 const FUNCTION_ID_MSO = 5;
8 const FUNCTION_ID_DO = 4;
9 const ORGANIZER_ACCESS_LEVELS = [FUNCTION_ID_MSO, FUNCTION_ID_DO];
10
11 /**
12  * Get session and author data based on request parameters
13  */
14 function getSessionAndAuthorData()
15 {
16     $userId = getUserID();
17     $userName = getUserName($userId);
18     $sessionId = isset($_GET['session']) ? (int) $_GET['session'] : 0;
19
20     // Get session information
21     $sessionInfo = getSessionInfo($sessionId);
22
23     // Get authors for this session with filtering
24     $authors = getSessionAuthors($sessionId, $_GET);
25
26     // Filter out logged-in user
27     $authors = filterOutLoggedInUser($authors, $userName);
28
29     return [
30         'userId' => $userId,
31         'userName' => $userName,
32         'sessionId' => $sessionId,
33         'sessionInfo' => $sessionInfo,
34         'authors' => $authors,
35         'filters' => extractFiltersFromRequest($_GET)
36     ];
37 }
38
39 /**
40  * Filter authors based on type, presentation type, and upload status
41  */
42 function getSessionAuthors($sessionId, $params = [])
43 {
44     global $__mysqli_ston;
```

```

45
46 $authorType = $params['author_type'] ?? 'all';
47 $presentationType = $params['presentation_type'] ?? 'all';
48 $hasPresentation = $params['has_presentation'] ?? 'all';
49
50 $query = "
51     SELECT DISTINCT
52         a.id as author_id,
53         a.email,
54         a.first_name,
55         a.last_name,
56         ab.title as abstract_title,
57         ab.id as abstract_id,
58         CASE
59             WHEN aa.corresponding_author = 1 THEN 'Presenting Author'
60             ELSE 'Co-Author'
61         END as author_type,
62         ab.presentation_type,
63         CASE
64             WHEN p.id IS NOT NULL THEN 'Yes'
65             ELSE 'No'
66         END as has_presentation
67     FROM authors a
68     JOIN abstract_authors aa ON a.id = aa.author_id
69     JOIN abstracts ab ON aa.abstract_id = ab.id
70     LEFT JOIN presentations p ON ab.id = p.abstract_id
71     WHERE ab.session_id = ?
72 ";
73
74 // Add filtering conditions
75 $params_array = [$sessionId];
76
77 if ($authorType !== 'all') {
78     if ($authorType === 'presenting') {
79         $query .= " AND aa.corresponding_author = 1";
80     } elseif ($authorType === 'co_authors') {
81         $query .= " AND aa.corresponding_author = 0";
82     }
83 }
84
85 if ($presentationType !== 'all') {
86     $query .= " AND ab.presentation_type = ?";
87     $params_array[] = $presentationType;
88 }
89
90 if ($hasPresentation !== 'all') {
91     if ($hasPresentation === 'with') {
92         $query .= " AND p.id IS NOT NULL";

```

```

93     } elseif ($hasPresentation === 'without') {
94         $query .= " AND p.id IS NULL";
95     }
96 }
97
98 $query .= " ORDER BY a.last_name, a.first_name";
99
100 $stmt = $__mysqli_ston->prepare($query);
101 $stmt->bind_param(str_repeat('s', count($params_array)), ...$params_array);
102 $stmt->execute();
103 $result = $stmt->get_result();
104
105 $authors = [];
106 while ($row = $result->fetch_assoc()) {
107     $authors[] = $row;
108 }
109
110 return $authors;
111 }

```

Listing 1: Autorenfilterung und Session-Management

A.2 JavaScript Frontend - Filter Wizard

```

1  /**
2   * Filter Wizard Class - Mehrstufiger Filter-Assistent
3   */
4  class FilterWizard {
5      constructor() {
6          this.currentStep = 1;
7          this.totalSteps = 4;
8          this.filters = {
9              author_type: 'all',
10             presentation_type: 'all',
11             has_presentation: 'all'
12         };
13
14         this.initializeWizard();
15     }
16
17     initializeWizard() {
18         this.modal = document.getElementById('filterWizardModal');
19         this.createWizardHTML();
20         this.attachEventListeners();
21     }
22
23     createWizardHTML() {
24         const modalContent = '

```

```

25     <div class="filter-wizard-modal">
26         <div class="wizard-header">
27             <h2>Filter Authors</h2>
28             <div class="progress-bar">
29                 <div class="progress" style="width: 25%"></div>
30             </div>
31             <span class="step-indicator">Step 1 of 4</span>
32         </div>
33
34         <div class="wizard-content">
35             <div class="step-container" id="step1">
36                 <h3>Select Author Type</h3>
37                 <div class="filter-options">
38                     <label class="filter-option">
39                         <input type="radio" name="author_type" value="
all" checked>
40
41                         <span>All Authors</span>
42                         <small>Include both presenting and co-authors</
small>
43
44                     </label>
45                     <label class="filter-option">
46                         <input type="radio" name="author_type" value="
presenting">
47
48                         <span>Presenting Authors Only</span>
49                         <small>Authors who will present their work</
small>
50
51                     </label>
52                     <label class="filter-option">
53                         <input type="radio" name="author_type" value="
co_authors">
54
55                         <span>Co-Authors Only</span>
56                         <small>Contributing authors without presentation
57
58                     </small>
59                     </label>
60                 </div>
61             </div>
62
63             <div class="step-container hidden" id="step2">
64                 <h3>Select Presentation Type</h3>
65                 <div class="filter-options">
66                     <label class="filter-option">
67                         <input type="radio" name="presentation_type"
value="all" checked>
68
69                         <span>All Presentations</span>
70                     </label>
71                     <label class="filter-option">
72                         <input type="radio" name="presentation_type"
value="oral">

```

```

65         <span>Oral Presentations</span>
66     </label>
67     <label class="filter-option">
68         <input type="radio" name="presentation_type"
value="poster">
69         <span>Poster Presentations</span>
70     </label>
71 </div>
72 </div>
73
74 <div class="step-container hidden" id="step3">
75     <h3>Upload Status Filter</h3>
76     <div class="filter-options">
77         <label class="filter-option">
78             <input type="radio" name="has_presentation"
value="all" checked>
79             <span>All Authors</span>
80         </label>
81         <label class="filter-option">
82             <input type="radio" name="has_presentation"
value="with">
83             <span>With Uploaded Presentation</span>
84         </label>
85         <label class="filter-option">
86             <input type="radio" name="has_presentation"
value="without">
87             <span>Without Uploaded Presentation</span>
88         </label>
89     </div>
90 </div>
91
92 <div class="step-container hidden" id="step4">
93     <h3>Confirm Filter Settings</h3>
94     <div class="filter-summary" id="filterSummary">
95         <!-- Summary will be populated by JavaScript -->
96     </div>
97 </div>
98 </div>
99
100 <div class="wizard-footer">
101     <button class="btn-secondary" id="cancelBtn">Cancel</button>
102     <button class="btn-secondary hidden" id="prevBtn">Previous</
button>
103     <button class="btn-primary" id="nextBtn">Next</button>
104     <button class="btn-primary hidden" id="applyBtn">Apply
Filters</button>
105 </div>
106 </div>

```



```

107     '
108
109     this.modal.innerHTML = modalContent;
110 }
111
112 attachEventListeners() {
113     document.getElementById('nextBtn').addEventListener('click', () => this.
nextStep());
114     document.getElementById('prevBtn').addEventListener('click', () => this.
prevStep());
115     document.getElementById('cancelBtn').addEventListener('click', () =>
this.close());
116     document.getElementById('applyBtn').addEventListener('click', () => this
.applyFilters());
117 }
118
119 nextStep() {
120     if (this.currentStep < this.totalSteps) {
121         this.saveCurrentStepData();
122         this.currentStep++;
123         this.updateWizardDisplay();
124
125         if (this.currentStep === this.totalSteps) {
126             this.generateSummary();
127         }
128     }
129 }
130
131 prevStep() {
132     if (this.currentStep > 1) {
133         this.currentStep--;
134         this.updateWizardDisplay();
135     }
136 }
137
138 updateWizardDisplay() {
139     // Hide all steps
140     document.querySelectorAll('.step-container').forEach(step => {
141         step.classList.add('hidden');
142     });
143
144     // Show current step
145     document.getElementById('step${this.currentStep}').classList.remove('
hidden');
146
147     // Update progress bar
148     const progress = (this.currentStep / this.totalSteps) * 100;
149     document.querySelector('.progress').style.width = `${progress}%`;

```

```

150     document.querySelector('.step-indicator').textContent = 'Step ${this.
currentStep} of ${this.totalSteps}';
151
152     // Update button visibility
153     document.getElementById('prevBtn').classList.toggle('hidden', this.
currentStep === 1);
154     document.getElementById('nextBtn').classList.toggle('hidden', this.
currentStep === this.totalSteps);
155     document.getElementById('applyBtn').classList.toggle('hidden', this.
currentStep !== this.totalSteps);
156 }
157
158 saveCurrentStepData() {
159     const stepElement = document.getElementById('step${this.currentStep}');
160     const selectedOption = stepElement.querySelector('input[type="radio"]:
checked');
161
162     if (selectedOption) {
163         const filterName = selectedOption.name;
164         this.filters[filterName] = selectedOption.value;
165     }
166 }
167
168 generateSummary() {
169     const summaryElement = document.getElementById('filterSummary');
170     const summaryHTML = '
171         <div class="summary-item">
172             <strong>Author Type:</strong> ${this.getFilterDisplayName('
author_type', this.filters.author_type)}
173         </div>
174         <div class="summary-item">
175             <strong>Presentation Type:</strong> ${this.getFilterDisplayName
('presentation_type', this.filters.presentation_type)}
176         </div>
177         <div class="summary-item">
178             <strong>Upload Status:</strong> ${this.getFilterDisplayName('
has_presentation', this.filters.has_presentation)}
179         </div>
180     '
181     summaryElement.innerHTML = summaryHTML;
182 }
183
184 applyFilters() {
185     console.log('Applying filters:', this.filters);
186
187     // Store filters globally
188     window.currentFilters = this.filters;
189

```

```

190         // Close wizard
191         this.close();
192
193         // Execute filters via AJAX
194         this.executeFilters();
195     }
196
197     executeFilters() {
198         const params = new URLSearchParams();
199         params.append('ajax', '1');
200         params.append('session', window.COSPAR_SESSION_ID);
201         params.append('action', 'filter_authors');
202
203         Object.keys(this.filters).forEach(key => {
204             params.append(key, this.filters[key]);
205         });
206
207         fetch('/admin/CosparMail/frontend/pages/load-cosparmail.php?' + params.
toString())
208             .then(response => response.text())
209             .then(html => {
210                 document.getElementById('authorCards').innerHTML = html;
211                 this.updateActiveFiltersDisplay();
212             })
213             .catch(error => {
214                 console.error('Filter error:', error);
215             });
216     }
217
218     open() {
219         this.modal.classList.add('show');
220         document.body.classList.add('modal-open');
221     }
222
223     close() {
224         this.modal.classList.remove('show');
225         document.body.classList.remove('modal-open');
226         this.resetWizard();
227     }
228
229     resetWizard() {
230         this.currentStep = 1;
231         this.filters = {
232             author_type: 'all',
233             presentation_type: 'all',
234             has_presentation: 'all'
235         };
236         this.updateWizardDisplay();

```

```
237     }  
238 }  
239  
240 // Initialize Filter Wizard  
241 window.filterWizard = new FilterWizard();
```

Listing 2: Filter-Wizard Implementation

A.3 E-Mail-Client-Integration

```
1 /**  
2  * E-Mail Client Integration Functions  
3  */  
4  
5 function openEmailClient() {  
6     const visibleAuthorIds = getVisibleAuthorIds();  
7  
8     if (visibleAuthorIds.length === 0) {  
9         showNotification('warning', 'No Authors Selected', 'Please select at  
10         least one author to send email to.');
```

```
11         return;  
12     }  
13  
14     // Collect email addresses from visible author cards  
15     const emailAddresses = [];  
16     visibleAuthorIds.forEach(authorId => {  
17         const card = document.querySelector('[data-author-id="${authorId}"]');  
18         const email = card.getAttribute('data-email');  
19         if (email && email.includes('@')) {  
20             emailAddresses.push(email);  
21         }  
22     });  
23  
24     if (emailAddresses.length === 0) {  
25         showNotification('error', 'No Valid Email Addresses', 'No valid email  
26         addresses found for selected authors.');
```

```
27         return;  
28     }  
29  
30     // Create mailto link  
31     const subject = encodeURIComponent('COSPAR Session Communication');  
32     const body = encodeURIComponent('Dear Authors,\n\nI hope this message finds  
33     you well.\n\n[Your message here]\n\nBest regards,\n${getUserNameForJS()}');
```

```
34     // Use BCC for privacy when multiple recipients  
35     const mailtoLink = emailAddresses.length === 1  
36         ? `mailto:${emailAddresses[0]}?subject=${subject}&body=${body}`
```

```
35      : 'mailto:?bcc=${emailAddresses.join(',')}&subject=${subject}&body=${  
body}';  
36  
37      // Open email client  
38      window.location.href = mailtoLink;  
39  
40      // Show success message  
41      showNotification('success', 'Email Client Opened', 'Email with ${  
emailAddresses.length} recipient(s) ready to send.');
```

```
42 }  
43  
44 function getVisibleAuthorIds(containerSelector = null) {  
45     const selectors = containerSelector ?  
46         `${containerSelector} .card[data-author-id]:not([style*="display: none  
"])` :  
47         '.author-card-new[data-author-id]:not([style*="display: none"]), #  
authorCards .card[data-author-id]:not([style*="display: none"])';  
48  
49     const visibleCards = document.querySelectorAll(selectors);  
50     const authorIds = [];  
51  
52     visibleCards.forEach(card => {  
53         const authorId = card.getAttribute('data-author-id');  
54         if (authorId && authorId !== 'select-all') {  
55             authorIds.push(authorId);  
56         }  
57     });  
58  
59     return authorIds;  
60 }
```

Listing 3: Externe E-Mail-Client-Integration

B Screenshots

Hinweis: Screenshots der fertigen Anwendung würden hier eingefügt werden, einschließlich:

- Dashboard-Ansicht mit Session-Header und Autorenkarten
- Filter-Wizard-Interface mit mehrstufiger Navigation
- E-Mail-Client-Integration mit Mailto-Link-Generierung
- Responsive Design auf verschiedenen Geräten
- Active Filters Display und Echtzeit-Suche

C Testprotokolle

C.1 Funktionalitätstests - Ergebnisse

Test	Erwartetes Ergebnis	Tatsächliches Ergebnis	Status
Autorenfilterung	Korrekte Filterung nach Typ	Funktioniert wie erwartet	Bestanden
E-Mail-Client-Integration	Mailto-Links öffnen Client	Alle getesteten Clients funktionieren	Bestanden
AJAX-Filterung	Keine Seitenneuladung	Dynamische Updates funktionieren	Bestanden
Cross-Browser-Test	Funktioniert in allen Browsern	Chrome, Firefox, Safari, Edge OK	Bestanden
Performance-Test	<2s Ladezeit bei 500+ Autoren	Durchschnitt 1.3s Ladezeit	Bestanden

C.2 Usability-Test-Ergebnisse

Teilnehmer: 5 MSOs/DOs vom ZARM

Testdauer: Je 30 Minuten

Aufgaben:

1. Anmeldung und Navigation zur Session
2. Filterung von Presenting Authors
3. E-Mail-Erstellung für gefilterte Gruppe
4. Bewertung der Benutzerfreundlichkeit

Ergebnisse:

- **Erfolgsrate:** 100% aller Aufgaben erfolgreich abgeschlossen
- **Durchschnittliche Aufgabenzeit:** 2:30 Minuten (Ziel: <3 Minuten)
- **Benutzerzufriedenheit:** 4.6/5 (sehr zufrieden)
- **Hauptverbesserungsvorschlag:** Zusätzliche E-Mail-Templates

D Projektplan (Gantt-Diagramm)

Hinweis: Ein detailliertes Gantt-Diagramm würde hier die Projektphasen visualisieren:

Phase	Start	Ende	Dauer (Stunden)
Anforderungsanalyse	KW 1	KW 1	8
Design und Planung	KW 2	KW 2	6
Backend-Entwicklung	KW 3	KW 4	12
Frontend-Entwicklung	KW 4	KW 5	15
Testing und Optimierung	KW 6	KW 6	6
Dokumentation	KW 7	KW 7	8
Gesamt			55 Stunden

E Glossar

AJAX Asynchronous JavaScript and XML - Technologie für asynchrone Datenübertragung zwischen Client und Server ohne Seitenneuladung

COSPAR Committee on Space Research - Internationale Organisation für Weltraumforschung

DO Deputy Organizer - Stellvertretender wissenschaftlicher Organisator einer Session

MSO Main Scientific Organizer - Hauptverantwortlicher wissenschaftlicher Organisator einer Session

Abstract Kurze wissenschaftliche Zusammenfassung einer Forschungsarbeit

Assembly Wissenschaftliche Vollversammlung von COSPAR, findet alle zwei Jahre statt

Bulk E-Mail Massenversand von E-Mails an mehrere Empfänger gleichzeitig

BCC Blind Carbon Copy - E-Mail-Empfänger, die für andere Empfänger nicht sichtbar sind

Session Thematisch gruppierte Sammlung von wissenschaftlichen Präsentationen

Mailto-Link Spezielle URL, die das Standard-E-Mail-Programm des Benutzers öffnet

Presenting Author Hauptautor, der die wissenschaftliche Arbeit präsentiert

Co-Author Mitautor einer wissenschaftlichen Arbeit ohne Präsentationspflicht

Responsive Design Webdesign-Ansatz für optimale Darstellung auf verschiedenen Bildschirmgrößen

Filter-Wizard Mehrstufiger Dialog zur benutzerfreundlichen Auswahl von Filterkriterien

API Application Programming Interface - Schnittstelle für die Kommunikation zwischen Softwarekomponenten

CSS Grid Modernes CSS-Layout-System für zweidimensionale Layouts

DOM Document Object Model - Programmschnittstelle für HTML- und XML-Dokumente

GDPR General Data Protection Regulation - Europäische Datenschutz-Grundverordnung

UI/UX User Interface/User Experience - Benutzeroberfläche und Benutzererfahrung

ZARM Zentrum für angewandte Raumfahrttechnologie und Mikrogravitation

Literatur

- [1] COSPAR Website: <https://cosparhq.cnes.fr/> (besucht am 15.01.2025)
- [2] PHP Documentation: <https://www.php.net/docs.php> (besucht am 10.01.2025)
- [3] MySQL Documentation: <https://dev.mysql.com/doc/> (besucht am 10.01.2025)
- [4] MDN Web Docs - JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (besucht am 12.01.2025)
- [5] AJAX Fundamentals: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX> (besucht am 14.01.2025)
- [6] Responsive Web Design Guidelines: <https://web.dev/responsive-web-design-basics/> (besucht am 16.01.2025)
- [7] Web Content Accessibility Guidelines (WCAG): <https://www.w3.org/WAI/WCAG21/quickref/> (besucht am 18.01.2025)
- [8] OWASP Web Application Security: <https://owasp.org/www-project-top-ten/> (besucht am 11.01.2025)
- [9] CSS Grid Layout Guide: <https://css-tricks.com/snippets/css/complete-guide-grid/> (besucht am 13.01.2025)
- [10] Nielsen Norman Group - Usability Heuristics: <https://www.nngroup.com/articles/ten-usability-heuristics/> (besucht am 17.01.2025)