|  |
| --- |
| IPA-Bericht Yannic Schüpbach |
| Dokumentenhistorie   |  |  |  |  | | --- | --- | --- | --- | | **Aktion** | **Datum** | **Autor** | **Änderung** | | Erstellen | 08.04.2024 | Yannic Schüpbach | Erstellen des Dokumentes | | Ergänzung | 08.04.2024 | Yannic Schüpbach | Analyse | | Ergänzung | 09.04.2024 | Yannic Schüpbach | Entwurf | |  |  |  |  | |

Inhaltsverzeichnis

1. Abbildungsverzeichnis 3

2. Kurzfassung 3

3. Vorwort 3

4. IPA Bericht Teil1 3

4.1 Aufgabenstellung 3

4.2 Ausgangslage 3

4.3 Tests 3

4.4 Dokumentationen 3

4.5 Vorkenntnisse 3

4.6 Vorarbeiten 3

4.7 Zeitplan 3

4.8 Projektaufbauorganisation 3

4.9 Arbeitsjournale 4

4.10 Zeitplan mit Ist-Soll 4

5. IPA Bericht Teil2 4

5.1 Planung 4

5.1.1 Analyse 5

5.1.2 Design 5

5.1.3 Implementierung 5

5.1.4 Testing 5

5.1.5 Dokumentation 5

5.2 Analyse der Aufgabenstellung 6

5.2.1 Ausgangslage 6

5.2.2 Systemübersicht 6

5.2.3 Sicherung der Arbeit 6

5.2.4 Abgrenzung 7

5.2.5 Lösungsansätze 7

5.3 Entwurf 8

5.3.1 Use-Case-Diagramm 8

5.3.2 Use-Case-Beschreibungen 9

5.3.3 GUI 14

5.3.4 Lösungsansätze 16

5.4 Programmcodedokumentation 16

6. Benutzerhandbuch 16

7. Glossar 16

8. Schlusswort 16

9. Quellen- und Literaturverzeichnis 16

10. Anhang 16

# Abbildungsverzeichnis

# Kurzfassung

### Kurze Ausgangssituation

Es wird eine Ablösung des bestehenden Telefonbuchs benötigt, welche moderner und angenehmer zu nutzen ist. Auf der Website soll es möglich sein Mitarbeiter anhand von Vornamen, Name, Mail, Telefonnummer, Abteilung, Funktion und Kostenstelle zu suchen. Sobald die gesuchte Person gefunden wurde, wird die Möglichkeit bestehen über das Telefonbuch diese Person zu kontaktieren.

### Umsetzung

### Ergebnis

# Vorwort

# IPA Bericht Teil1

## Aufgabenstellung

Die Aufgabe des Projekts besteht darin eine Webapplikation zu machen, in welcher die Mitarbeiter des USB andere Mitarbeiter anhand von verschiedenen Suchkriterien ausfindig machen können. Diese Suchkriterien sind: Vorname, Name, Telefonnummer, Mail, Abteilung, Funktion und Kostenstelle. Ebenfalls sollen die Mitarbeiter den gesuchten Mitarbeiter kontaktieren können. Dafür soll in der Applikation die Möglichkeit bestehen über einen Klick auf die Telefonnummer, Mail und Teams, den entsprechenden Mitarbeiter mit einem Anruf, einer Mail oder einer Teams Nachricht zu kontaktieren.

## Ausgangslage

Das USB verfügt bereits über ein Telefonbuch, jedoch ist dieses aus 2007 und veraltet. Die Oberfläche bietet verschiedene Suchfelder für die Suchkriterien und das Design entspricht nicht mehr der den Standards des USB. Das bestehende Telefonbuch bietet nicht die Möglichkeit nach der Mail zu suchen. Es besteht auch nicht die Möglichkeit die gefundenen Mitarbeiter mit einem Klick auf die Telefonnummer, Mail oder Teams zu kontaktieren. Die Datenquelle für meine Applikation und die bestehende wird die Gleiche sein. Mithilfe von LDAP können alle gewünschten Daten abgefragt und gefunden werden. In LDAP sind alle Mitarbeiter und deren Daten, welche für meine Applikation relevant sind, gespeichert.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

## Tests

## Dokumentationen

## Vorkenntnisse

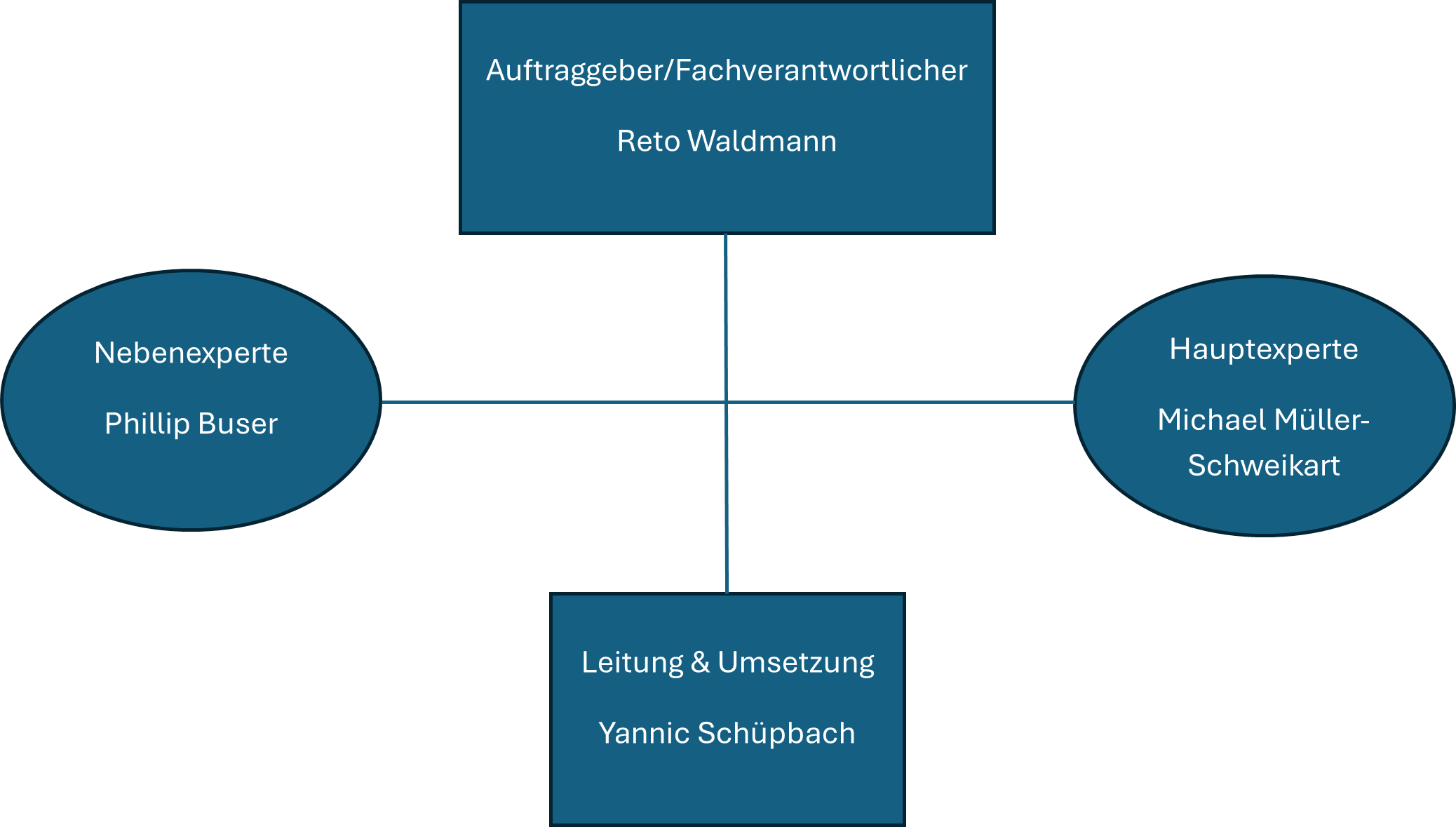
## Vorarbeiten

Damit ich einfach mit der Arbeit starten kann habe ich diverse Vorarbeiten durchgeführt. Im Vorhinein wurde der LDAP-Zugriff beantragt. Diesen habe ich erhalten. Ich habe meine PHP, HTML und CSS-Kenntnisse aufgefrischt damit wieder alles aktuell ist. Ich habe mich über LDAP-Abfragen und Verbindung informiert.

Die Anforderungen für das Projekt wurden definiert und Fragen geklärt.

Das ist der erstellte Zeitplan ohne Soll ist Vergleich.

## Projektaufbauorganisation



Die Projektorganisation besteht aus einem Auftraggeber, dem Fachexperten Reto Waldmann, den Experten und der Leitung und Umsetzung, welche von Yannic Schüpbach repräsentiert wird. Die Projektaufbauorganisation stellt sicher, dass alle wichtigen Rollen besetzt sind und eine effiziente Zusammenarbeit und Kommunikation ermöglicht wird.

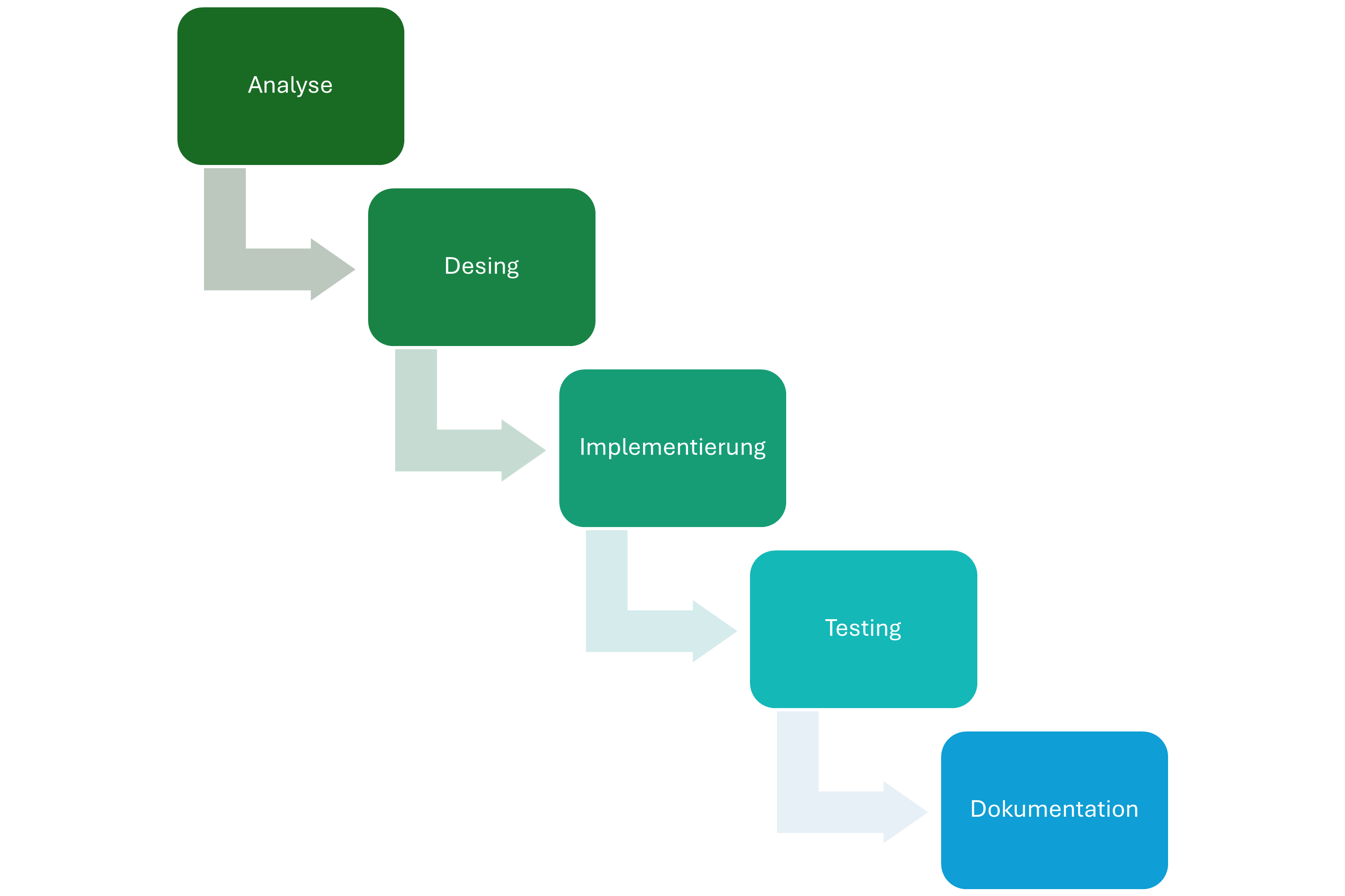
## Arbeitsjournale

## Zeitplan mit Ist-Soll

# IPA Bericht Teil2

## Planung

Da eine IPA mit einer Dauer von 10 Tagen überschaubar ist und ich allein arbeite habe ich mich für das Phasenmodell entschieden. Es stellt sich als die Optimale Wahl für mich herausgestellt, weil sie das Projekt in klar definierte und aufeinanderfolgende Phasen unterteilt. Dies erleichtert die Planung und Überwachung des Fortschritts erheblich. Deshalb habe ich meine IPA in fünf Phasen unterteilt.



### Analyse

In dieser Phase wird die Arbeit geplant und analysiert. Der Zeitplan wird erstellt und die Methode wird verankert. Die Projektorganisiation wird beschrieben und es werden mehrere Optionen analysiert und es wird sich für den Ansatz entschieden, welcher umgesetzt wird.

### Design

In dieser Phase wird ein Gerüst aus der Analyse ausgearbeitet. Dies umfasst die Gestaltung des GUI’s, die Erstellung eines Use Case Diagramms inklusive Beschreibung der Datenquelle.

### Implementierung

In dieser Phase wird die Applikation anhand der Analyse und des Designs programmiert. Die Phase endet mit der Fertigstellung der Applikation, welche dann anschliessend getestet werden kann.

### Testing

In dieser Phase wird ein Testkonzept geschrieben, mit welchem die Applikation anschliessen getestet wird. Die Ergebnisse der Tests werden in einem Testprotokoll festgehalten.

### Dokumentation

In dieser Phase wird der IPA Bericht geschrieben. Das Benutzerhandbuch wird erstellt und Dokumente werden zusammengefügt. Die Programmdokumentation und die Arbeitsjournale werden eingefügt. Anschliessend wir das ganze korrigiert.

## Analyse der Aufgabenstellung

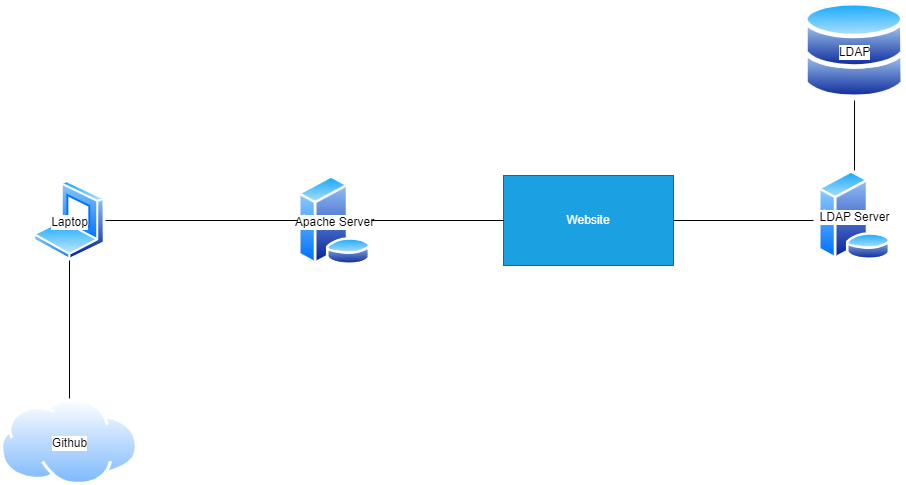
Die Aufgabe des Projekts besteht darin eine Webapplikation zu machen, in welcher die Mitarbeiter des USB andere Mitarbeiter anhand von verschiedenen Suchkriterien ausfindig machen können. Diese Suchkriterien sind: Vorname, Name, Telefonnummer, Mail, Abteilung, Funktion und Kostenstelle. Ebenfalls sollen die Mitarbeiter den gesuchten Mitarbeiter kontaktieren können. Dafür soll in der Applikation die Möglichkeit bestehen über einen klick auf die Telefonnummer, Mail und Teams, den entsprechenden Mitarbeiter mit einem Anruf, einer Mail oder einer Teams Nachricht zu kontaktieren.

Aus dieser Aufgabenstellung kristallisieren sich zwei grosse Ziele heraus. Die Suche nach Mitarbeitern anhand verschiedener Kriterien, damit auch Mitarbeiter gefunden werden können, von welchen andere Kriterien wie Name oder Vorname nicht bekannt sind. Das zweite grosse Ziel ist die Kontaktaufnahme mit den gefundenen Mitarbeitern. Dafür muss eine Anbindung an Outlook, Teams und Webex erstellt werden.

### Ausgangslage

Das USB verfügt bereits über ein Telefonbuch, jedoch ist dieses aus 2007 und veraltet. Die Oberfläche bietet verschiedene Suchfelder für die Suchkriterien und das Design entspricht nicht mehr der den Standards des USB. Die Liste der Abteilungen ist zusätzlich hardcoded. Das bestehende Telefonbuch bietet nicht die Möglichkeit nach der Mail zu suchen. Es besteht auch nicht die Möglichkeit die gefundenen Mitarbeiter mit einem klick auf die Telefonnummer, Mail oder Teams zu kontaktieren. Die Datenquelle für meine Applikation und die bestehende wird die Gleiche sein. Mithilfe von LDAP können alle gewünschten Daten abgefragt und gefunden werden. In LDAP sind alle Mitarbeiter und deren Daten, welche für meine Applikation relevant sind, gespeichert.

### Systemübersicht



In dieser Abbildung ist das System beschrieben. Die Applikation wird auf dem Laptop mithilfe von XAMPP auf einen Apache Server gehostet. Die Applikation baut eine Verbindung zum LDAP-Server auf und frag bei diesem Daten ab. Der LDAP-Server gibt dann mit den Daten aus LDAP die entsprechenden Ergebnisse zurück. Die Applikation verwendet PHP, HTML, CSS und wird in Visual Studio Code programmiert. Die Applikation wird mithilfe von Github gesichert.

### Sicherung der Arbeit

Die Sicherung des Programms und des IPA-Berichts ist ein essenzieller Teil der Arbeit um Datenverlust und Arbeitsunterbrechungen zu vermeiden. Damit eine kontinuierliche und zuverlässige Sicherung der Arbeit zu gewährleisten, werden Github und OneDrive verwendet. Das Programm und der IPA-Bericht werden jeden Abend auf Github hochgeladen. GitHub bietet eine effiziente Möglichkeit zur sicheren Versionsverwaltung und schützt die Arbeit zuverlässig vor unbeabsichtigten Änderungen oder Verlusten. Die weit verbreitete und vertrauenswürdige Plattform ermöglicht nicht nur den jederzeitigen Zugriff auf frühere Dateiversionen, sondern erlaubt auch die übersichtliche Nachverfolgung von Änderungen, was einen enormen Vorteil bei der Fehlerbehebung bietet. Als zusätzliche Sicherung werden alle Dateien im persönlichen OneDrive-Ordner des Unispitals abgelegt.

### Abgrenzung

Die Webapplikation wird lokal gehostet, das Hochladen auf einen Server ist kein Bestandteil der Arbeit. Das Benutzer Handbuch wird nur für USB-Mitarbeiter geschrieben.

### Lösungsansätze

Für die Darstellung des GUI in der Webapplikation gibt es mehrere Möglichkeiten.

**1. Option**

Ich erstelle ein Wireframe für die Webapplikation. Ich habe bereits bei der Validierung ein Wireframe erstellt. Für diese Option würde ich das Wireframe nochmals überarbeiten und erweitern. Wireframes sind grundlegende, schematische Darstellungen von Webapplikationen oder Apps, sie visualisieren den Aufbau und die Struktur der Applikation. Wireframes erhalten keine Details des Designs und Farben. Das macht Wireframes zu der am wenigsten aufwändige Option.

**2. Option**

Ich designe ein Mockup der Webapplikation. Dieses ist der Applikation sehr ähnlich ist jedoch nicht interaktiv. Dadurch hat man eine Vorstellung wie das fertige Produkt aussehen soll. Mockups enthalten grundlegende Designelemente, Farben, Schriftarten, Icons etc. Dadurch liefern sie ein realistisches Bild der Applikation ohne das Entwicklungsressourcen auf die Entwicklung eines funktionsfähigen Prototyps verwendet werden müssen.

**3. Option**

Ich entwickle einen interaktiven Prototyp der Applikation. Dadurch wird ein realistisches Bild des fertigen Programms geboten. Prototypen sind interaktive Versionen von Applikation, welche es ermöglichen die Struktur, das Benutzererlebnis und die Funktionalität zu visualisieren. Prototypen sind meist mit hohem Aufwand verbunden.

**Entscheidung**

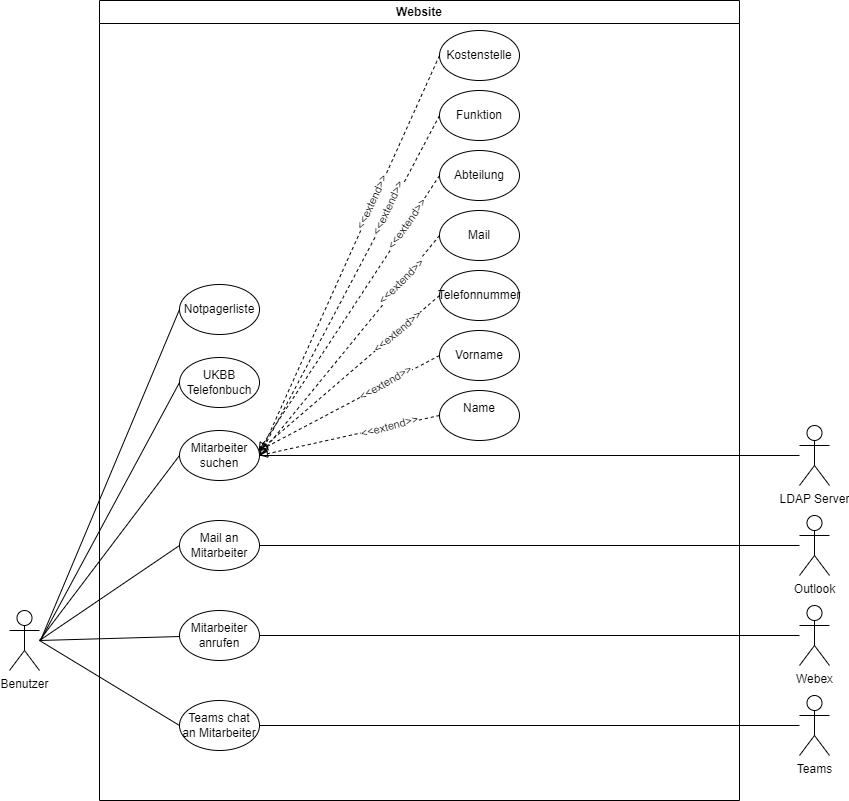
Ich habe mich für die zweite Option, das Erstellen von einem Mockup, entschieden. Der Grund dafür ist, dass Mockups detaillierter sind als Wireframes aber bei weitem nicht so aufwendig sind als ein Prototyp, welcher alle Funktionen bereits visualisiert. Da die Applikation, vor der Entwicklung, nicht präsentiert und verkauft werden muss, ist ein Prototyp nicht erforderlich. Mockups sind in dieser Arbeit ausreichend, um die Designziele zu erreichen und das GUI zu entwerfen, dass spart wesentliche Ressourcen in der Designphase.

## Entwurf

Folgende Entwürfe werden für die Applikation verwendet:

* Use-Case-Diagramm
* Use-Case-Beschreibung
* GUI

### Use-Case-Diagramm



### Use-Case-Beschreibungen

**Mitarbeiter suchen**

|  |  |
| --- | --- |
| Ziel | Akteur kann im Suchfeld nach Mitarbeitern suchen. |
| Akteure | USB-Mitarbeiter |
| Auslösendes Ereignis | Ausfüllen des Suchfelds  Akteur löst die Suche aus |
| Vorbedingung | Akteur befindet sich im Intranet  Akteur hat das Suchfeld ausgefüllt |
| Beschreibung | 1. Suchfeld ausfüllen 2. «Suchen» Knopf drücken oder «Enter» Taste 3. Eingabe wird geprüft 4. Resultat wird angezeigt |
| Alternativszenario | Fehlermeldung das keine Einträge zu den Suchkriterien gefunden wurden. |
| Nachbedingung | Suche wurde durchgeführt und dem Benutzer werden die Resultate angezeigt. |

**Mail**

|  |  |
| --- | --- |
| Ziel | Akteur kann über einen klick eine neue Mail aus der Applikation starten. |
| Akteure | USB-Mitarbeiter |
| Auslösendes Ereignis | Akteur klickt auf die Mailadresse in einem Eintrag. |
| Vorbedingung | Akteur hat eine Suche durchgeführt  Resultate werden angezeigt |
| Beschreibung | 1. Auf Mailadresse in einem Eintrag klicken 2. Outlook öffnet sich 3. Eine neue Outlookmail wird geöffnet 4. Die angeklickte Mailadresse wird in Outlook eingefügt |
| Alternativszenario | Keine Mail im Eintrag vorhanden. |
| Nachbedingung | - |

**Anrufen**

|  |  |
| --- | --- |
| Ziel | Akteur kann über einen klick einen Anruf aus der Applikation starten. |
| Akteure | USB-Mitarbeiter |
| Auslösendes Ereignis | Akteur klickt auf die Telefonnummer in einem Eintrag. |
| Vorbedingung | Akteur hat eine Suche durchgeführt  Resultate werden angezeigt |
| Beschreibung | 1. Auf Telefonnummer in einem Eintrag klicken 2. Webex öffnet sich 3. Webex fragt, ob der Anruf gestartet werden soll. 4. Klick auf «Ja» 5. Anruf wird mit der Nummer aus dem Eintrag gestartet |
| Alternativszenario | Keine Telefonnummer im Eintrag vorhanden. |
| Nachbedingung | - |

**Teams Chat**

|  |  |
| --- | --- |
| Ziel | Akteur kann über einen klick einen neuen Teams Chat aus der Applikation öffnen |
| Akteure | USB-Mitarbeiter |
| Auslösendes Ereignis | Akteur klickt auf «Open with Teams» im Eintrag |
| Vorbedingung | Akteur hat eine Suche durchgeführt  Resultate werden angezeigt |
| Beschreibung | 1. Auf «Open with Teams» im Eintrag klicken 2. Teams öffnet sich 3. Eine neuer Teams Chat wird mit der Person aus dem Eintrag geöffnet |
| Alternativszenario | Keine Mail im Eintrag vorhanden, heisst Teams Chat öffnen nicht möglich. |
| Nachbedingung | - |

**UKBB Telefonbuch**

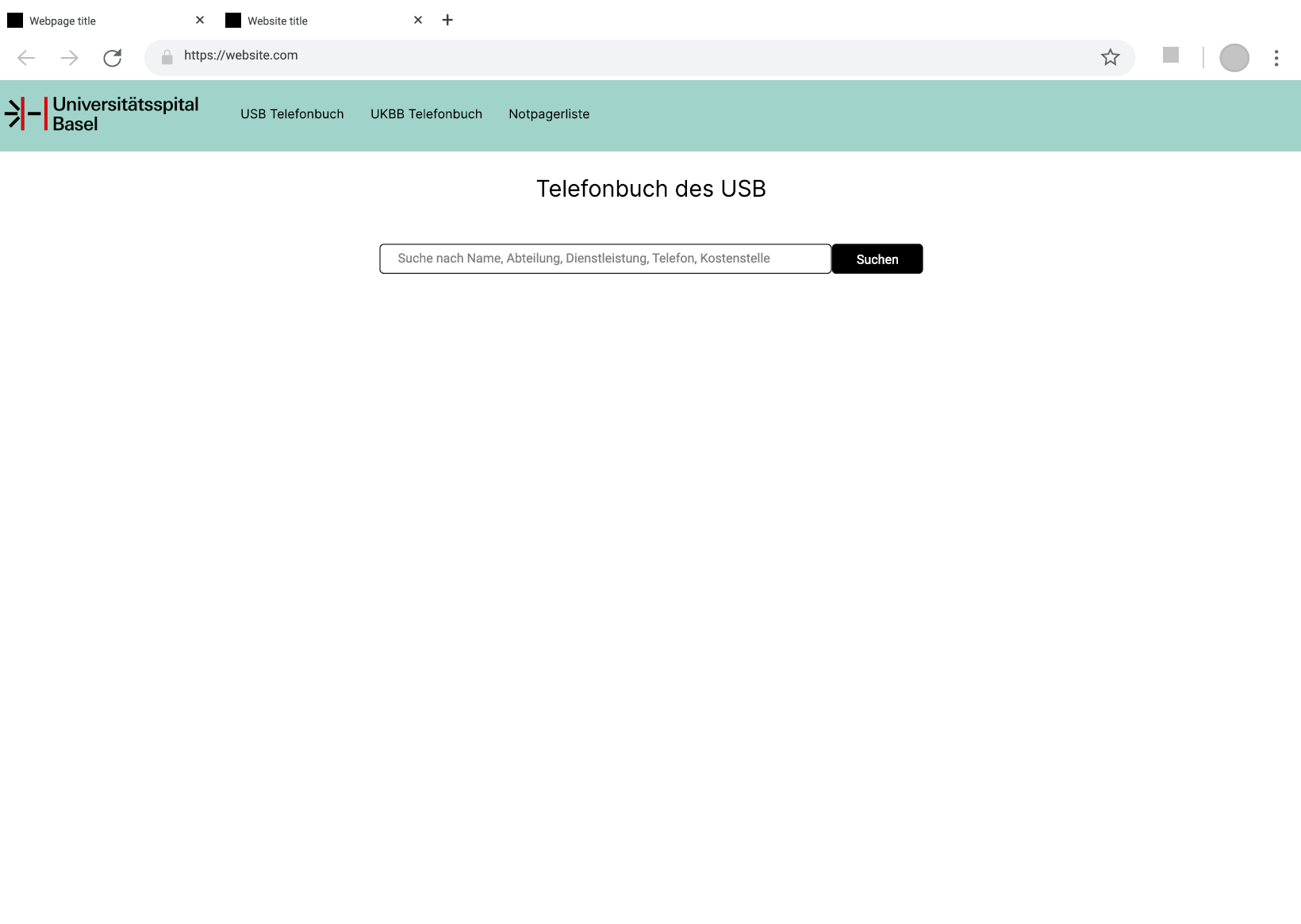
|  |  |
| --- | --- |
| Ziel | Akteur kann über einen klick auf das Telefonbuch des UKBB gelangen |
| Akteure | USB-Mitarbeiter |
| Auslösendes Ereignis | Akteur klickt auf «UKBB Telefonbuch» |
| Vorbedingung | Akteur ist im Intranet |
| Beschreibung | 1. Auf «UKBB Telefonbuch» klicken 2. Ein neuer Tab öffnet sich mit dem UKBB-Telefonbuch |
| Alternativszenario | Das UKBB-Telefonbuch ist nicht verfügbar |
| Nachbedingung | - |

**Notpagerliste**

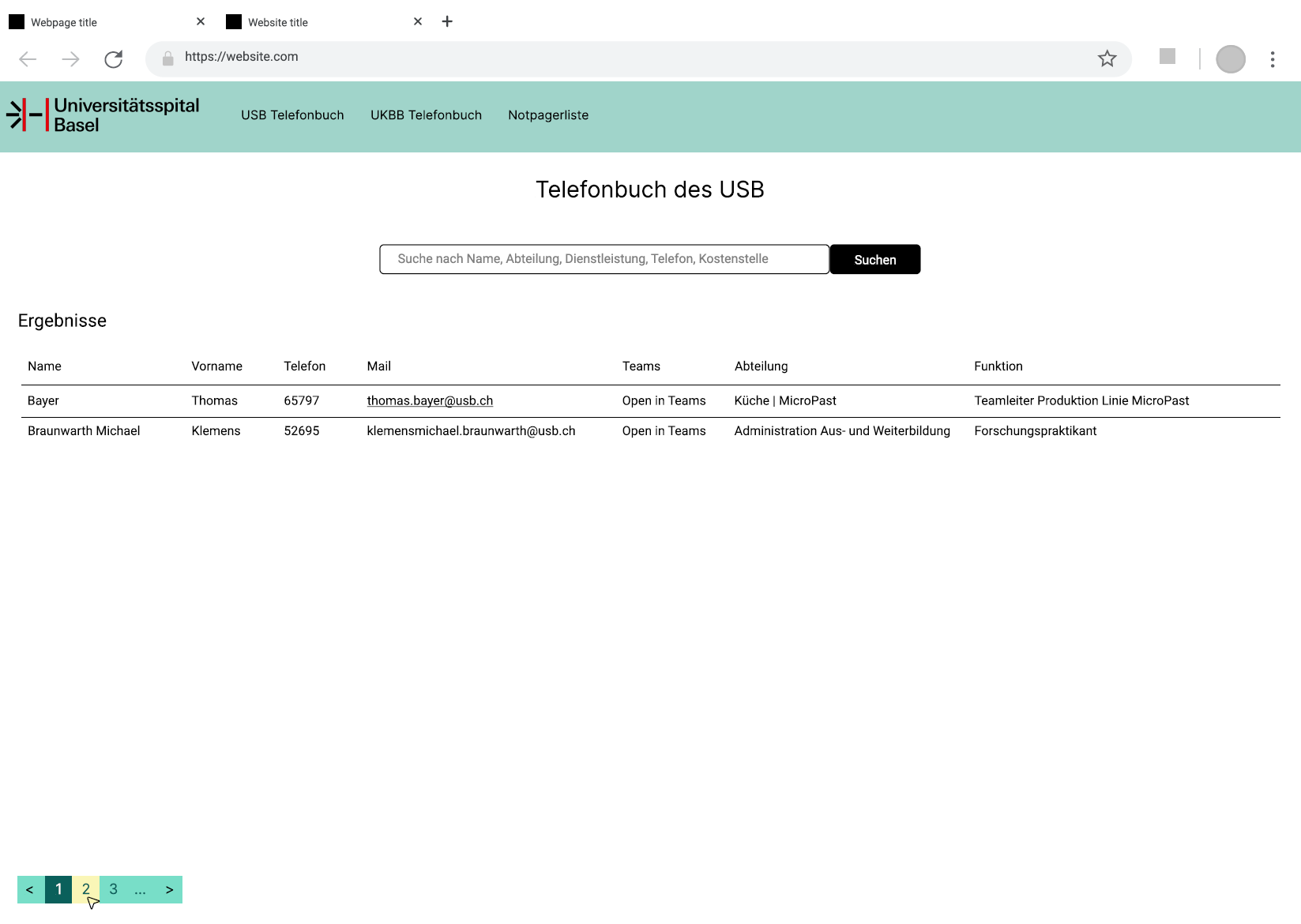
|  |  |
| --- | --- |
| Ziel | Akteur kann über einen klick auf die Notpagerliste gelangen. |
| Akteure | USB-Mitarbeiter |
| Auslösendes Ereignis | Akteur klickt auf «Notpagerliste» |
| Vorbedingung | Akteur ist im Intranet |
| Beschreibung | 1. Auf «Notpagerliste» klicken 2. Ein neuer Tab öffnet sich mit der Notpagerliste |
| Alternativszenario | - |
| Nachbedingung | - |

### GUI

**Suchen**



**Ergebnisse anzeigen**



### Lösungsansätze

Es gibt verschiedene Optionen wie die Applikation umgesetzt und gestaltet wird. Ich habe mir verschiedene Optionen angeschaut und daraus einen Lösungsansatz erarbeitet.

**1.Option**

Ich gestalte meine Applikation aus komplett eigenem CSS und HTML und verzichte auf jegliche Frameworks. Das würde den Vorteil bieten, dass ich jeden Aspekt des Designs und des GUIS genau so gestalten kann wie ich will. Diese Option währe sehr aufwendig und würde viel Arbeit und Zeit in Anspruch nehmen, aufgrund der zeitlichen Beschränkung ist diese Option nicht für das Projekt geeignet.

**2. Option**

Ich verwende Frameworks wie zum Beispiel Bootstrap um meine Applikation zu gestalten. Bootstrap bietet viele Vorlagen und Layouts. Das würde viel Arbeit sparen und den Design Prozess einfacher machen. Das Problem wäre jedoch das Bootstrap nicht sehr flexibel ist. Die Farben welche vom USB für Webseiten vorgegeben werden unterstützt Bootstrap nicht. Auch wenn das Grid System von Bootstrap sehr nützlich ist, ist es sehr unflexibel. Dadurch müsste das Design wesentlich angepasst werden und ich könnte mich nicht an die Vorgaben vom USB halten.

**3. Option**

Ich verwende eine Mischung aus Bootstrap und eigenem CSS. Mit dieser Option ist es möglich mit Bootstrap den Grundriss der Applikation zu entwerfen. Für Designelemente, die den Vorschriften des USB unterliegen und mit Bootstrap nicht einfach zu machen sind, kann das eigene CSS-Stylesheet eingesetzt werden. Diese Option bietet die grösste Flexibilität und ist zeitlich umsetzbar.

**Entscheidung**

Ich habe mich für die dritte Option, eine Mischung aus Bootstrap und eigenem CSS entschieden. Es bietet die meiste Flexibilität und der Zeitaufwand wäre machbar. Bei dieser Option kann ich mich an die Vorschriften des USB halten, ein flexibles Design erstellen und mit Bootstrap Zeit sparen, wo es möglich ist.

## Programmcodedokumentation

In der Programmcodedokumentation werden die Highlights des Codes mithilfe von Bildern dargestellt und erklärt.

### LDAP-Verbindung

Die LDAP-Verbindung wurde mithilfe von zwei Dateien namens «ldap\_config.php» und «ldap\_connect.php» programmiert. In ldap\_config.php sind die LDAP-Zugangsdaten definiert und hinterlegt. In ldap\_connect.php werden die Variablen von der config abgerufen. Danach wird die Verbindung zum Server über $ldapConn und $ldapBind hergestellt.

In ldap\_connect.php wird mithilfe der Funktionen logLdapActivity und ldapErrorHandler ein neues Log File (ldap.log) erstellt. logLdapActivity wird in $ldapBind aufgerufen, wenn die Verbindung zum LDAP-Server erfolgreich ist. LdapErrorHandler wird ausgeführt so bald im Skript ein LDAP-Error auftritt das Ganze wird dann in das Log File geschrieben.

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

LDAP-Zugangsdaten -ldap\_config-php

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

LDAP-Verbindung -ldap\_connect.php

# Benutzerhandbuch

# Glossar

# Schlusswort

# Quellen- und Literaturverzeichnis

# Anhang