Requirements Document

# Übersicht

Dieses Dokument beschreibt die Anforderungen und Funktionalitäten des PMS (Patient Management System). Es zeigt eine externe Sicht in Form eines Überblickes was das PMS abdecken soll.

**Versionskontrolle**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Aktion** | **Beteiligte Personen** | **Datum** |
| 0.1 | Dokument initiiert | JH, AA, NS, AS, CB & TM | 09.04.18 |
| 0.2 | Gliederung erstellt und Layout optimiert | MH | 13.04.18 |
| 1.0 | Alle Kapitel ergänzt und zusammengetragen, Dokument finalisiert | MH | 15.04.18 |

## Zielgruppe

Das Dokument richtet sich an folgende Leserschaft und beschreibt wozu sie das Dokument verwenden:

* **Ärzte:** Spezifizieren die Anforderungen und validieren diese.
* **Kunden:** Spezifizieren Änderungen an den Anforderungen.
* **Manager:** Verwenden das Anforderungsdokument um ein Angebot für das System zu planen und den Systementwicklungsprozess zu planen.
* **Systemingenieure:** Nutzen die Anforderungen um zu verstehen, welches System entwickelt werden soll.
* **Systemtestingenieure:** Nutzen die Anforderungen um Validierungstests für das System zu entwickeln.
* **Systemwartungsingenieure:** Nutzen die Anforderungen um das System und die Beziehungen zwischen seinen Teilen zu verstehen.

Inhalt

[Übersicht 1](#_Toc511592983)

[Zielgruppe 1](#_Toc511592984)

[Tabellenverzeichnis 3](#_Toc511592985)

[Abbildungsverzeichnis 3](#_Toc511592986)

[Glossar 4](#_Toc511592987)

[Einführung 5](#_Toc511592988)

[User Requirements Definition 6](#_Toc511592989)

[Use Case Diagramm & User Requirements 6](#_Toc511592990)

[Graphische Benutzeroberfläche 7](#_Toc511592991)

[Beschreibung Use Case 2 13](#_Toc511592992)

[Ablauf 13](#_Toc511592993)

[Beschreibung Use Case 5 13](#_Toc511592994)

[Ablauf 14](#_Toc511592995)

[System Architecture 14](#_Toc511592996)

[Anforderungen System Architecture 15](#_Toc511592997)

[Anforderungen an Clients & Endgeräte 15](#_Toc511592998)

[Anforderungen an Server 15](#_Toc511592999)

[Anforderungen an Datenbank 15](#_Toc511593000)

[System Requirements Specification 15](#_Toc511593001)

[Funktionale Requirements 15](#_Toc511593002)

[Notiz zu Patient 15](#_Toc511593003)

[Report 15](#_Toc511593004)

[Medikamente 15](#_Toc511593005)

[Patientenakte 16](#_Toc511593006)

[Agenda 16](#_Toc511593007)

[Klinik 16](#_Toc511593008)

[Suchtverzeichnis 16](#_Toc511593009)

[Nicht-funktionale Requirements 16](#_Toc511593010)

[Usability 17](#_Toc511593011)

[Performance 17](#_Toc511593012)

[Sicherheit 17](#_Toc511593013)

[System Models 17](#_Toc511593014)

[System Evolution 18](#_Toc511593015)

[Testing 18](#_Toc511593016)

# Tabellenverzeichnis

[Tabelle 1 – Glossar 4](#_Toc511593030)

[Tabelle 2 - Beschreibung Use Case 2 13](#_Toc511593031)

[Tabelle 3 - Ablauf Use Case 2 13](#_Toc511593032)

[Tabelle 4 - Beschreibung Use Case 5 14](#_Toc511593033)

[Tabelle 5 - Ablauf Use Case 5 14](#_Toc511593034)

[Tabelle 6 - Ablauf User Request 18](#_Toc511593035)

# Abbildungsverzeichnis

[Abbildung 1 - Use Case Diagramm 6](#_Toc511593017)

[Abbildung 2 – Anmeldefenster 8](#_Toc511593018)

[Abbildung 3 – Hauptmenu 8](#_Toc511593019)

[Abbildung 4 - Use Case 1 - Notiz hinzufügen 9](#_Toc511593020)

[Abbildung 5 - Use Case 2 - Report generieren 9](#_Toc511593021)

[Abbildung 6 - Use Case 3 - Medikament suchen/verschreiben 10](#_Toc511593022)

[Abbildung 7 - Use Case 4 - Patientenakte suchen 10](#_Toc511593023)

[Abbildung 8 - Use Case 5 - Agenda mit Terminerfassung 11](#_Toc511593024)

[Abbildung 9 - Use Case 5 - Agenda mit Terminerfassung (neuer Termin) 11](#_Toc511593025)

[Abbildung 10 - Use Case 6 - Klinik suchen/anfragen 12](#_Toc511593026)

[Abbildung 11 - Use Case 7 - Sucht suchen/zuweisen 12](#_Toc511593027)

[Abbildung 12 - System Architecture 14](#_Toc511593028)

[Abbildung 13 - System Models 17](#_Toc511593029)

# Glossar

In diesem Abschnitt werden die technischen Begriffe sowie Abkürzungen beschrieben.

|  |  |
| --- | --- |
| **Abkürzung** | **Bedeutung** |
| AA | Aleks Andrejic |
| AM | Antonio Musardo |
| AS | Amin Schaller |
| CB | Cedric Bielmann |
| CSS | Cascading Style Sheets, Programmiersprache um Webapplikationen zu gestalten |
| HTML | Hypertext Markup Language  Programmiersprache um Webapplikationen zu strukturieren |
| HTTP | Hypertext Transfer Protocol, Technologie um Webseiten miteinander zu verknüpfen |
| HTTPS | Hypertext Transfer Protocol Secure, Technologie um Webseiten sicher miteinander zu verknüpfen |
| Java SE | Java Standard Edition, Programmiersprache |
| JH | Jan Henzi |
| MH | Michael Hofer |
| NS | Nicola Schlup |
| PMS | Patient Management System |
| RAID | Redundant Array of Independent Disks, Technologie um Speichersysteme vor Ausfällen zu schützen |

Tabelle – Glossar

# Einführung

Das PMS ist eine Applikation zur Unterstützung von Ärzten, welche sich hauptsächlich mit der Bestimmung und Behandlung von Suchtkrankheiten beschäftigen. Das zu erstellende System bietet folgende Grundfunktionalitäten:

* Agenda
* Suchtverzeichnis
* Medikamentenverzeichnis
* Anstaltsverzeichnis
* Reports

Die Agenda verwaltet sämtliche Termine eines Arztes. Sie bietet eine Wochenansicht zur Übersicht aller bevorstehenden Termine der aktuellen Arbeitswoche. Der Arzt kann Termine ansehen, bearbeiten, löschen und hinzufügen. Jeder Termin enthält diverse Informationen.

Das Suchtverzeichnis dient als eine Art Nachschlagewerk. Der Benutzer des Systems greift über eine Suche effizient auf Einträge des Verzeichnisses zu und erhält ausführliche Informationen wie Beschreibung, Symptome oder Behandlung zur ausgewählten Sucht.

Sollte der zu Patient medikamentös behandelt werden können, steht ein Medikamentenverzeichnis zur Verfügung. Es listet sämtliche angebotenen Medikamente und zeigt zu jedem Medikament die notwendigen Informationen wie beispielsweise Nebenwirkungen oder Einnahmedosen bzw. -zeitraum.

Da es vorkommen kann, dass der Patient nicht im aktuellen Spital behandelt werden kann, bietet das PMS eine Übersicht sämtlicher Gesundheitsanstalten inkl. Kontaktinformationen an. Patienten, welchen eine Therapie verordnet wird oder einen Suchtspezialisten besuchen sollten, können der entsprechenden Anstalt übergeben werden.

Über die Reports stehen individuell konfigurierbare Auswertungen zur Verfügung. Über diese ist es möglich, unter einer Anzahl an Filtern die verschiedenen Datenbanken abzufragen und Zusammenhänge herzustellen.

# User Requirements Definition

Das System bietet dem Arzt diverse Funktionen um ihm den Alltag zu erleichtern und zu vereinfachen. Im nachfolgenden Diagramm sind die Hauptfunktionen, die das System beinhalten soll ersichtlich.

## Use Case Diagramm & User Requirements

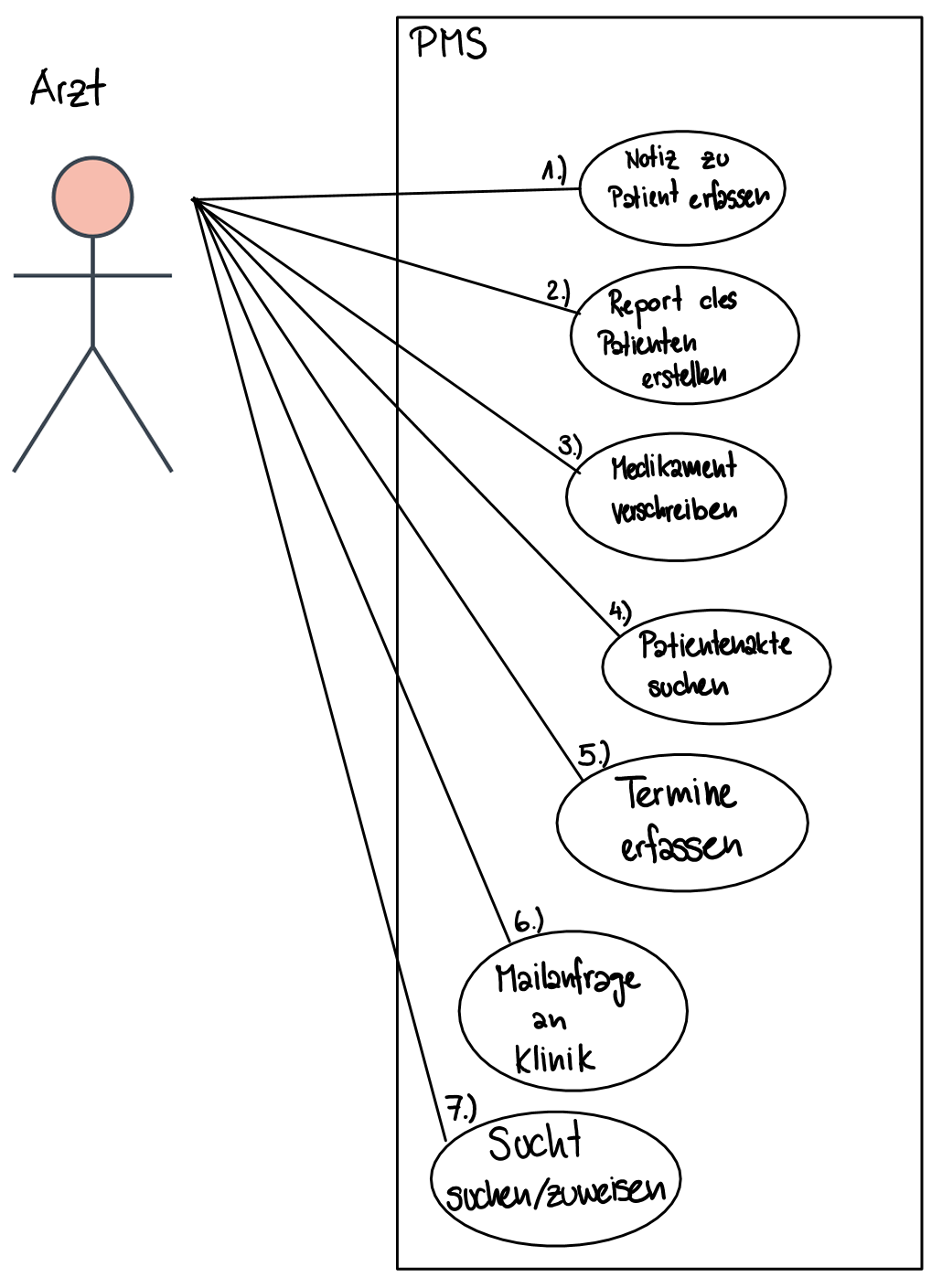


Abbildung - Use Case Diagramm

1. Notiz hinzufügen  
   Während den Sitzungen kann der Arzt in der entsprechenden Patientenakte seine persönlichen Notizen hinzufügen um Beobachtungen und weitere wichtige Informationen festzuhalten.
2. Report generieren  
   Bei Bedarf nach einer kompakten Übersicht zu diversen Informationen können verschiedene Reports generiert werden. Im Report enthalten sind alle relevanten Eigenschaften der Patienten und Kliniken.
3. Medikament suchen/verschreiben  
   Da es für eine Suchtbehandlung essentiell ist Medikamente verschreiben zu können, wird es dem Arzt durch das Patienten Management System ermöglicht einem Patienten diese direkt zuzuordnen. Dazu gehört auch eine Suchfunktion, die ein effizientes Finden der Medikamente mit Stichwörtern ermöglicht.
4. Patientenakte suchen  
   Zum Patienten Management System gehört auch die Suchfunktion der Patientenakte. Mit den verfügbaren Informationen kann eine Suche ausgelöst werden, die schliesslich alle passenden Ergebnisse liefert.
5. Agenda mit Terminerfassung  
   In der Patientenakte kann der Arzt seine Termine mit dem jeweiligen Patienten sehen. In der Kalenderfunktion findet sich eine kompakte Wochenansicht mit allen anstehenden Terminen wider. Darin kann er auch neue Termine erfassen.
6. Klinik suchen/anfragen  
   Um die Therapie des Patienten bestmöglich zu gestalten gibt es eine Möglichkeit nach Kliniken zu suchen und diesen per Mail eine Anfrage zukommen zu lassen.
7. Sucht suchen/zuweisen  
   Um eine richtige Diagnose machen zu können hat der Arzt die Möglichkeit nach Süchten zu suchen. Hat er die entsprechende gefunden kann diese direkt einer Patientenakte hinzugefügt werden.

## Graphische Benutzeroberfläche

Nachfolgen ist die graphische Benutzeroberfläche des Patienten Management Systems aufgeführt. Die Applikation wird danach aufgebaut wobei kleine Änderungen vorbehalten sind. In den ersten zwei Abbildungen findet sich die Anmeldung und das Hauptmenu der Applikation wider. Danach sind die Oberflächen zu den entsprechenden Use Cases abgebildet.

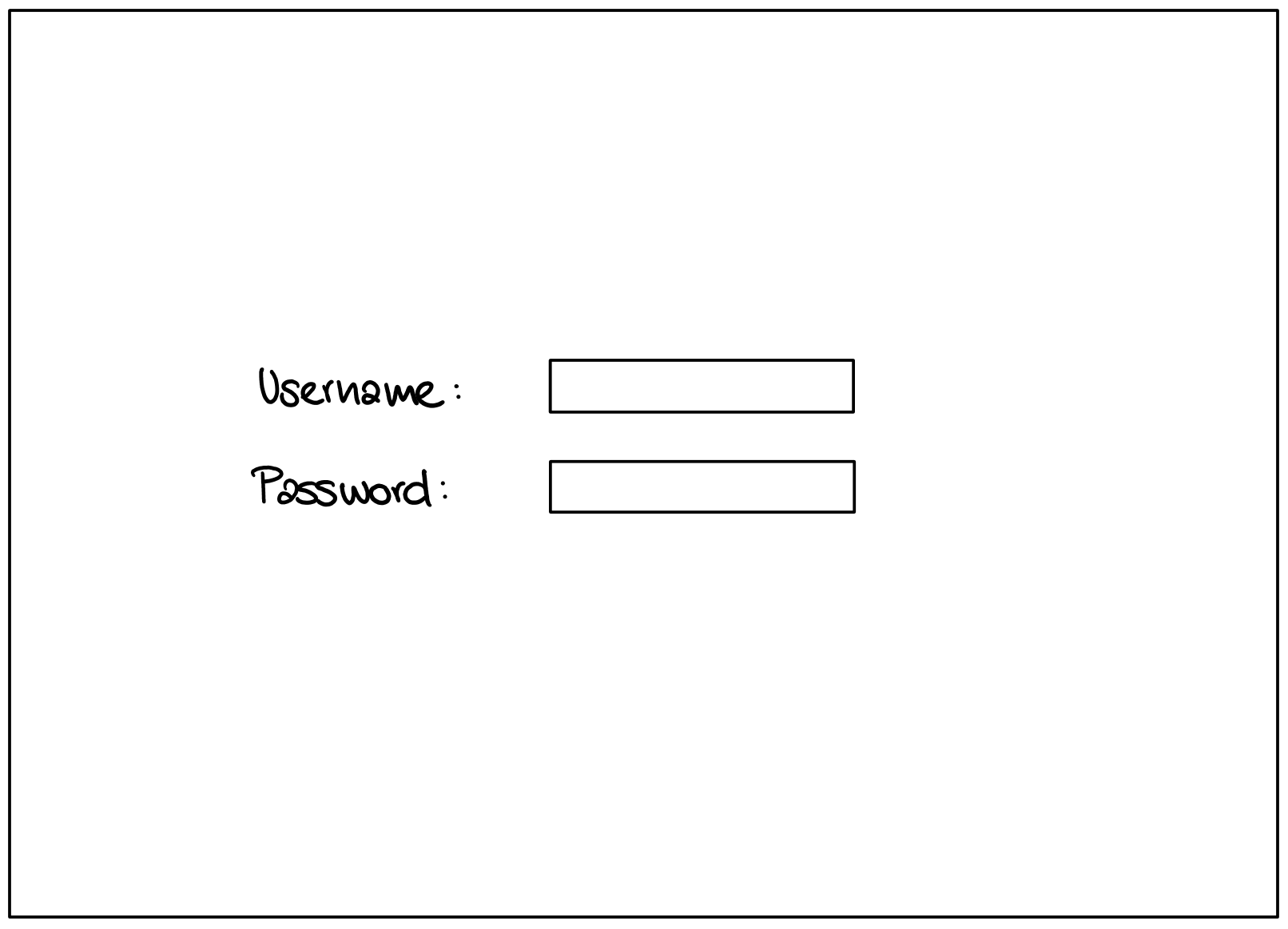


Abbildung – Anmeldefenster

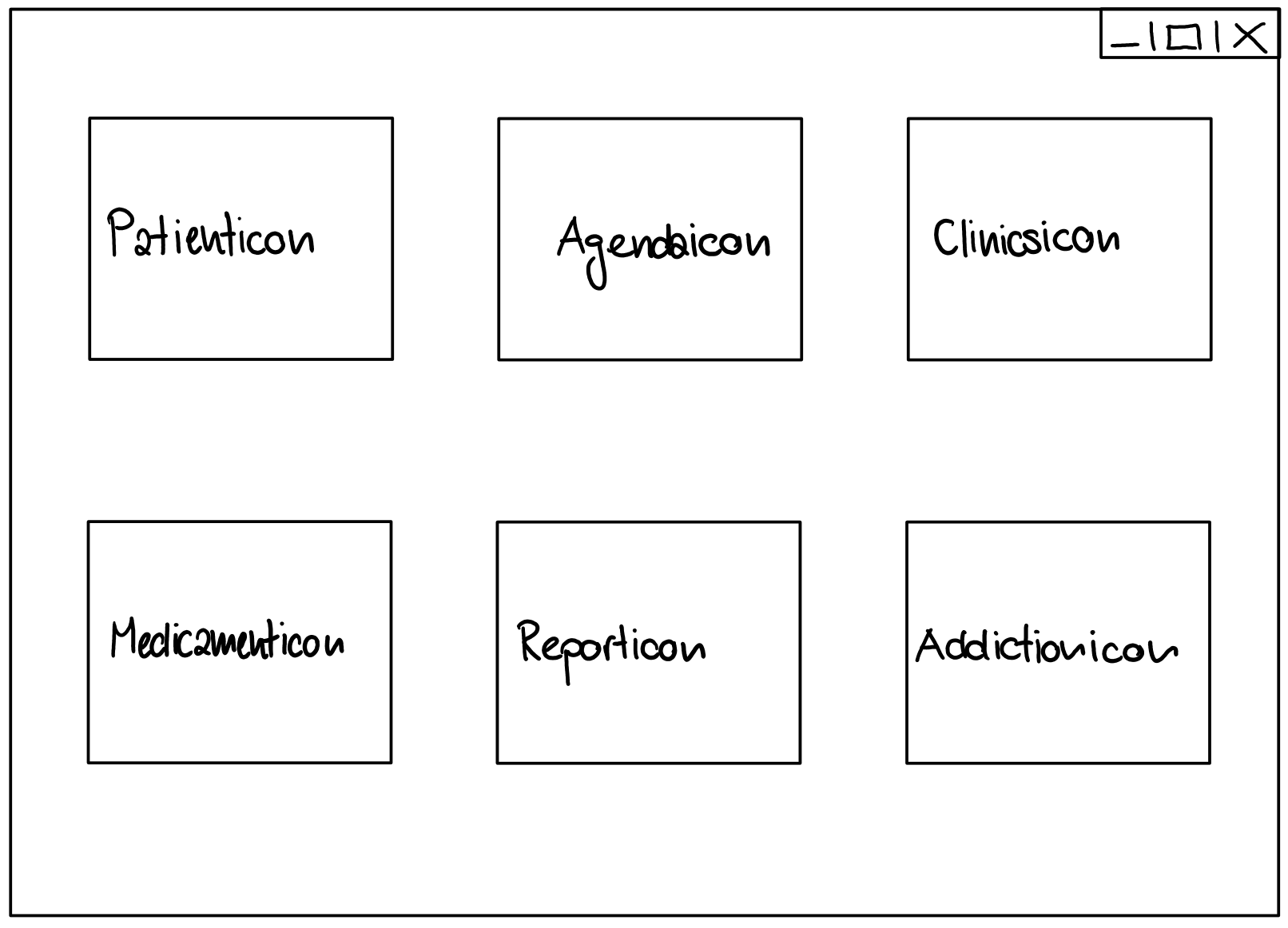


Abbildung – Hauptmenu

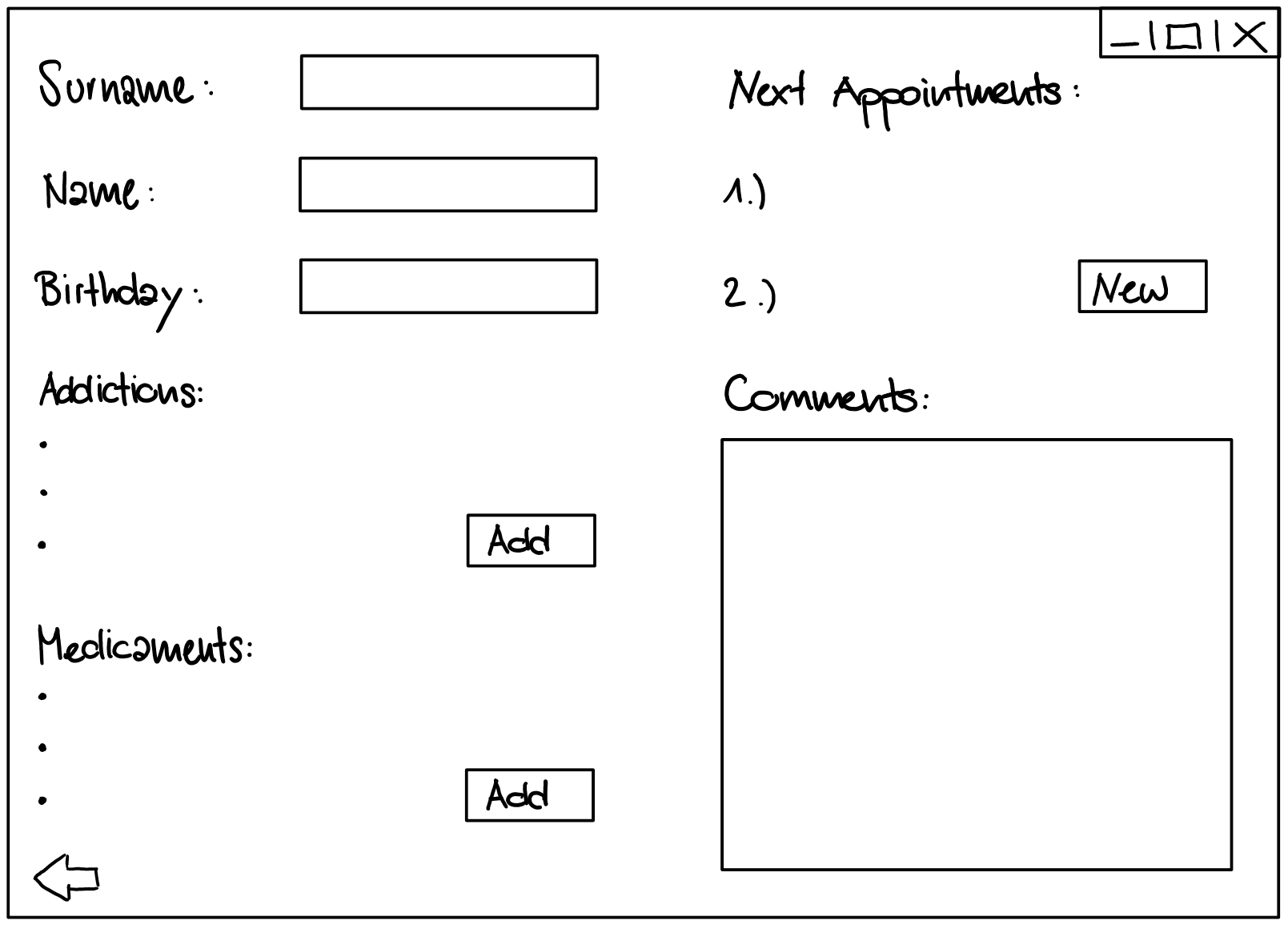


Abbildung - Use Case 1 - Notiz hinzufügen

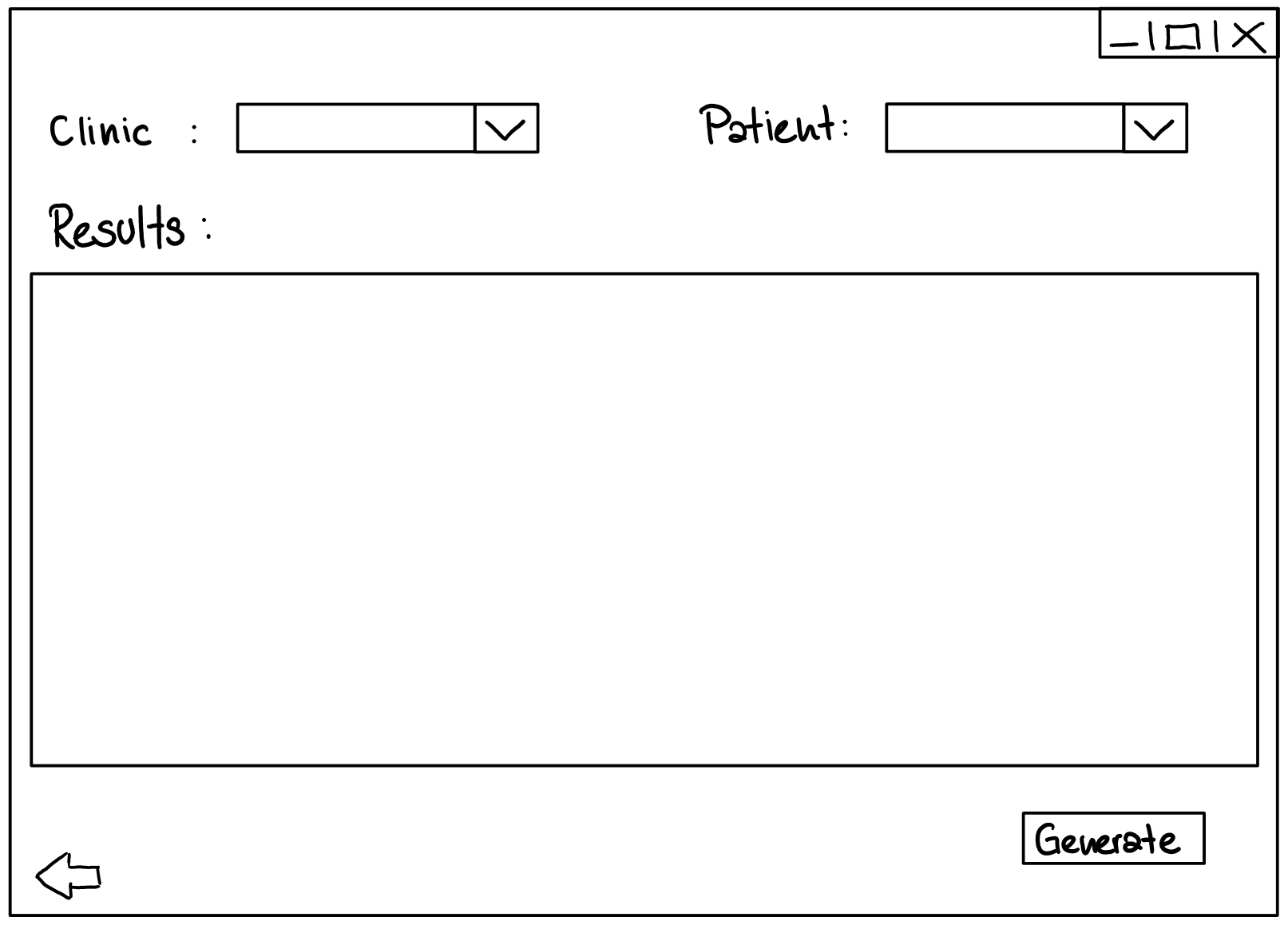


Abbildung - Use Case 2 - Report generieren

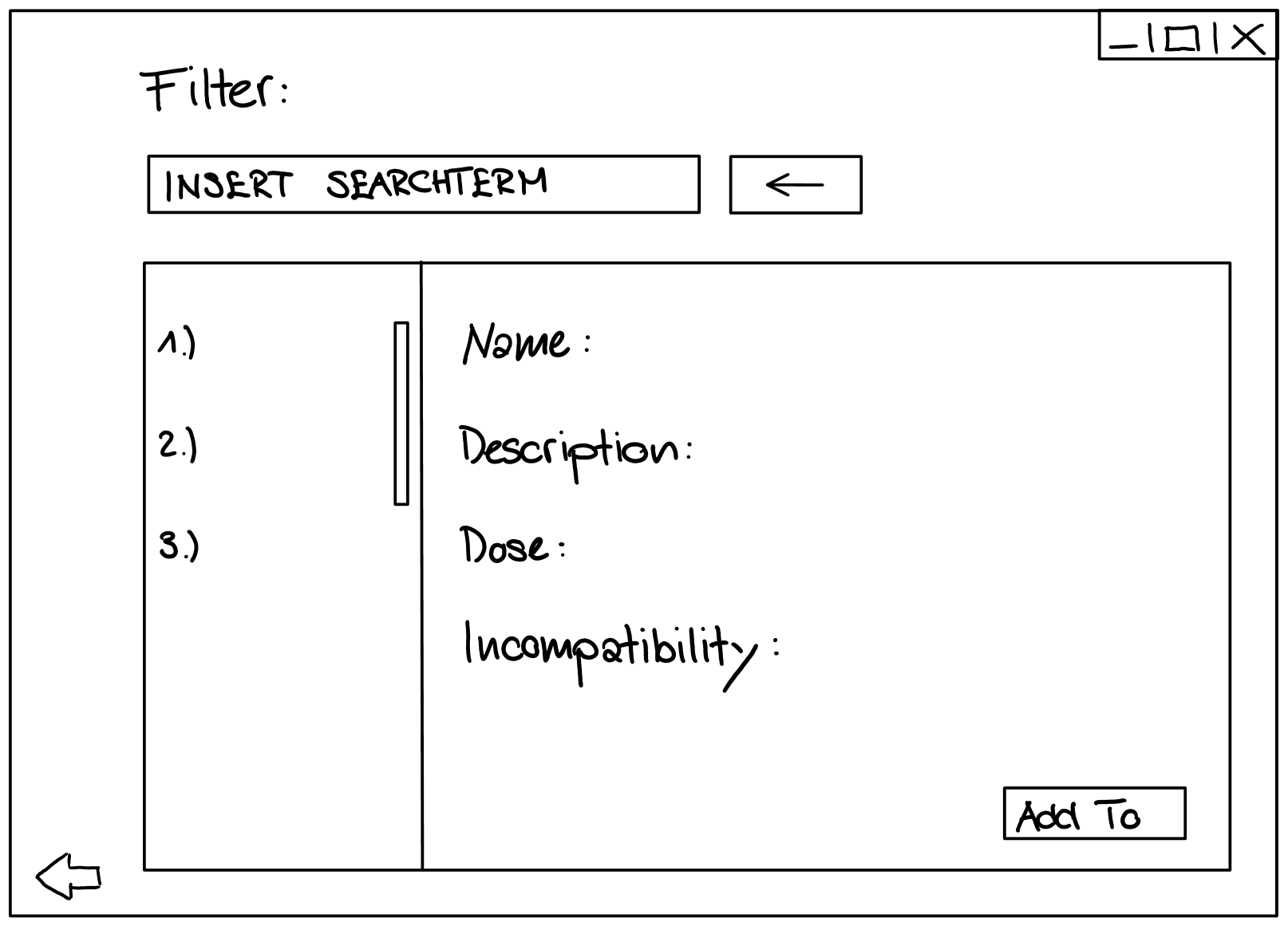


Abbildung - Use Case 3 - Medikament suchen/verschreiben

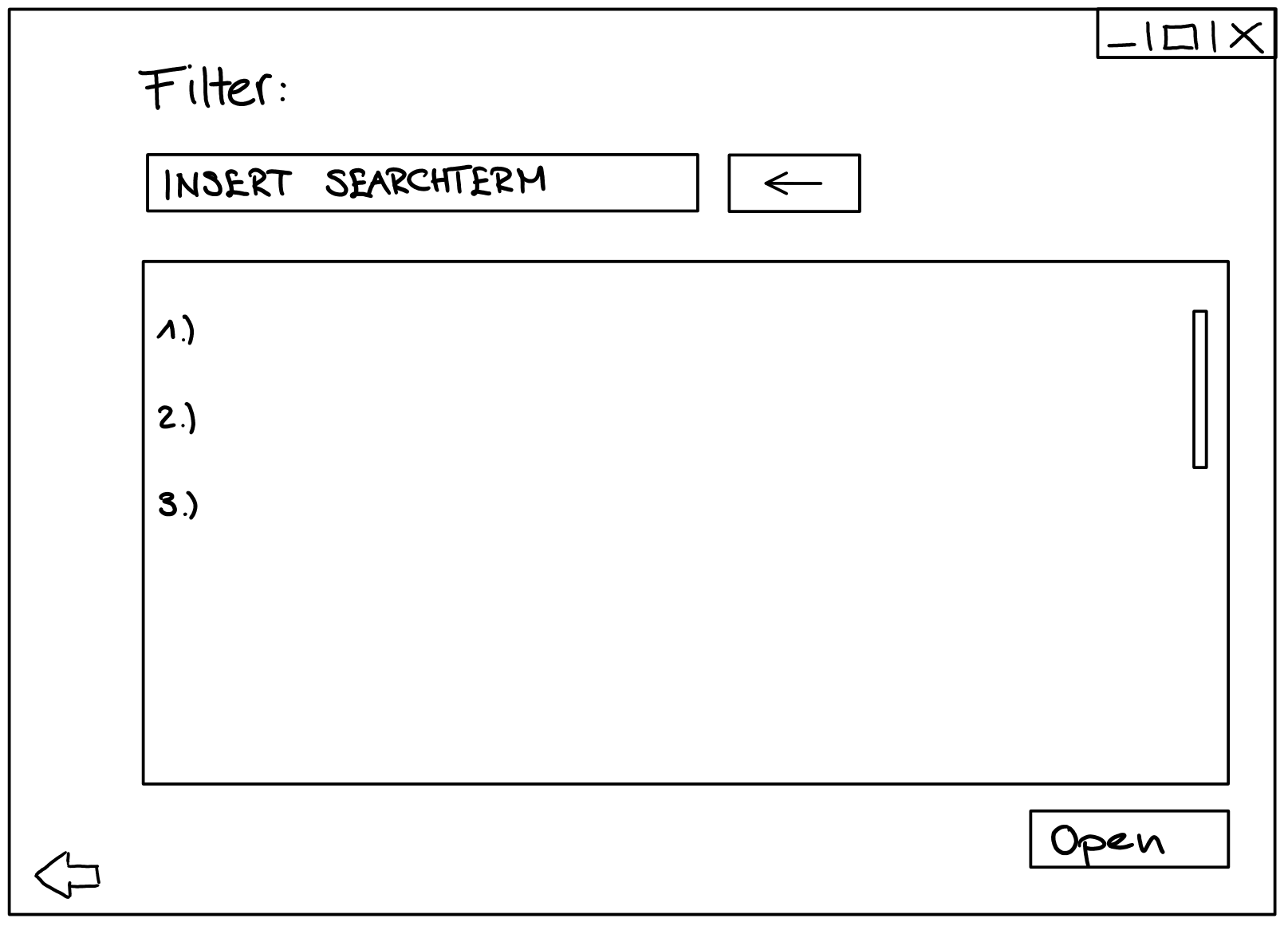


Abbildung - Use Case 4 - Patientenakte suchen

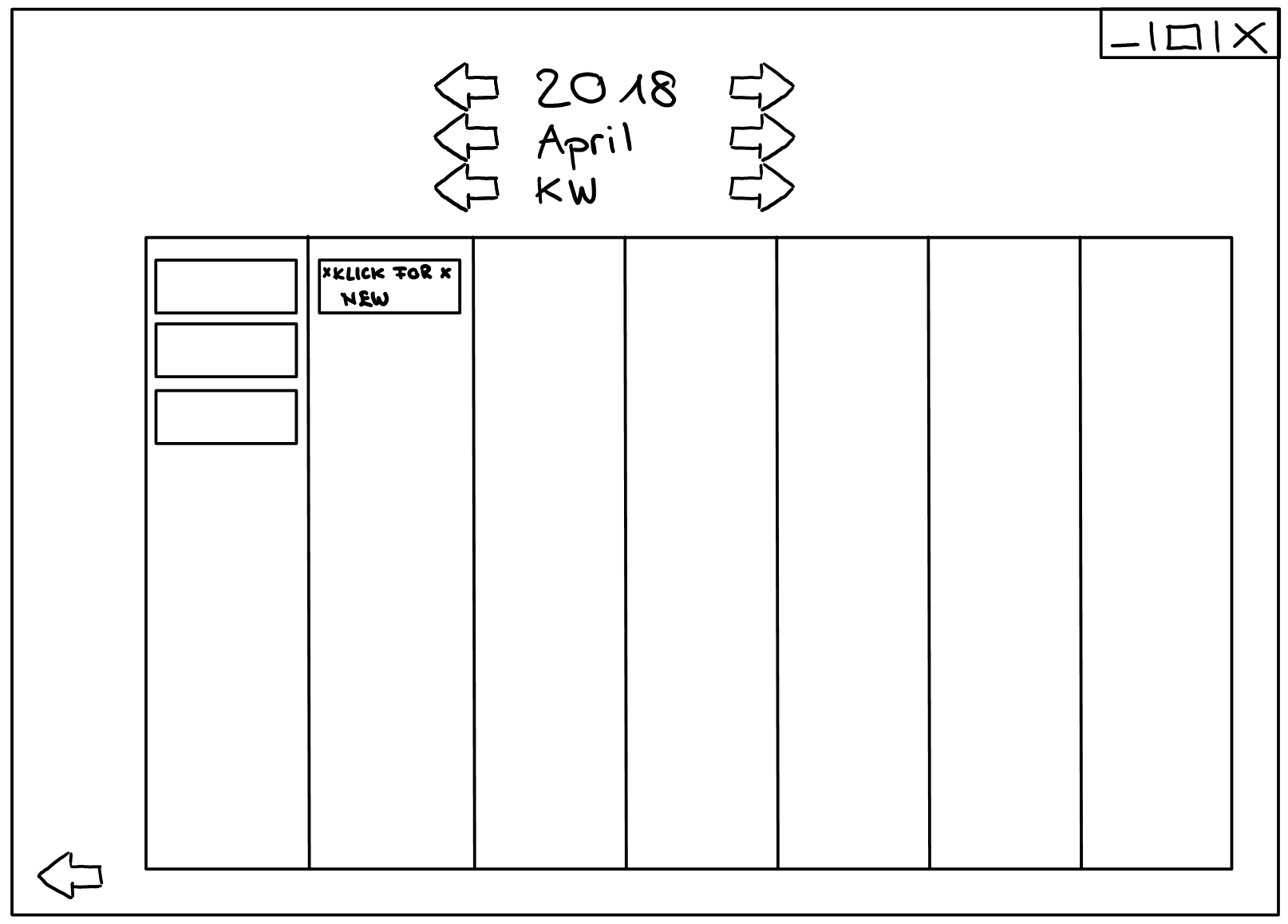


Abbildung - Use Case 5 - Agenda mit Terminerfassung

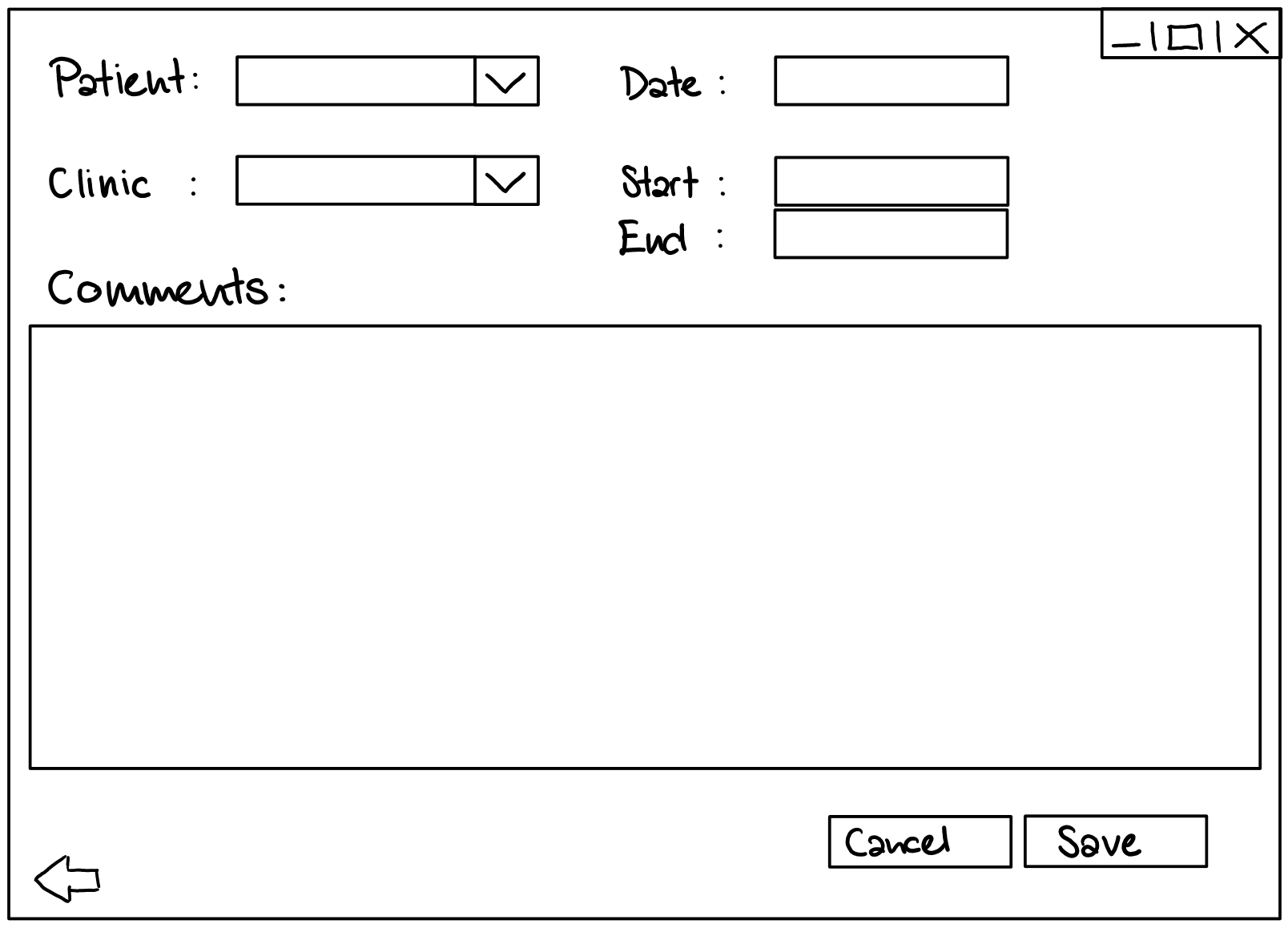


Abbildung - Use Case 5 - Agenda mit Terminerfassung (neuer Termin)

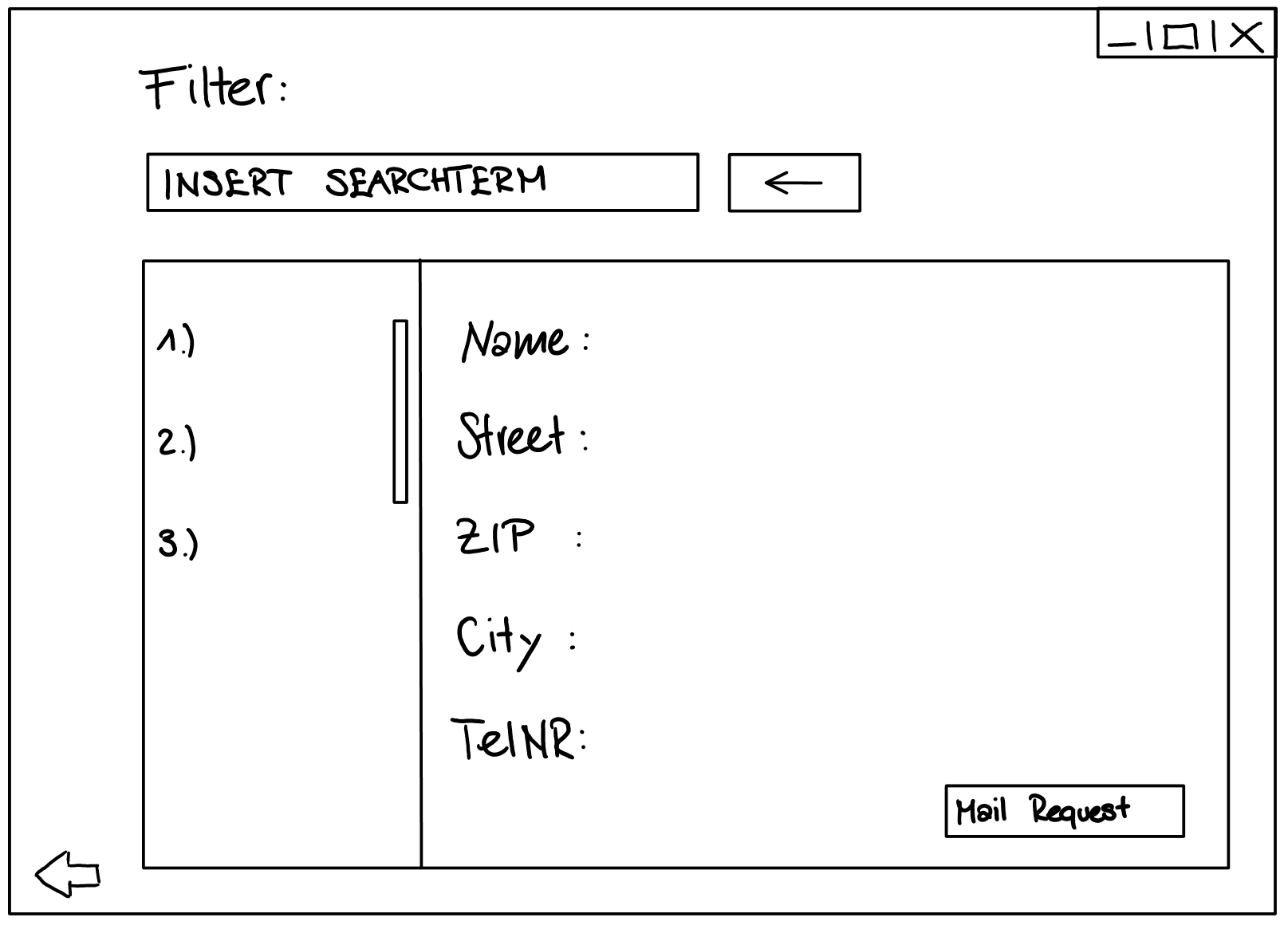


Abbildung - Use Case 6 - Klinik suchen/anfragen

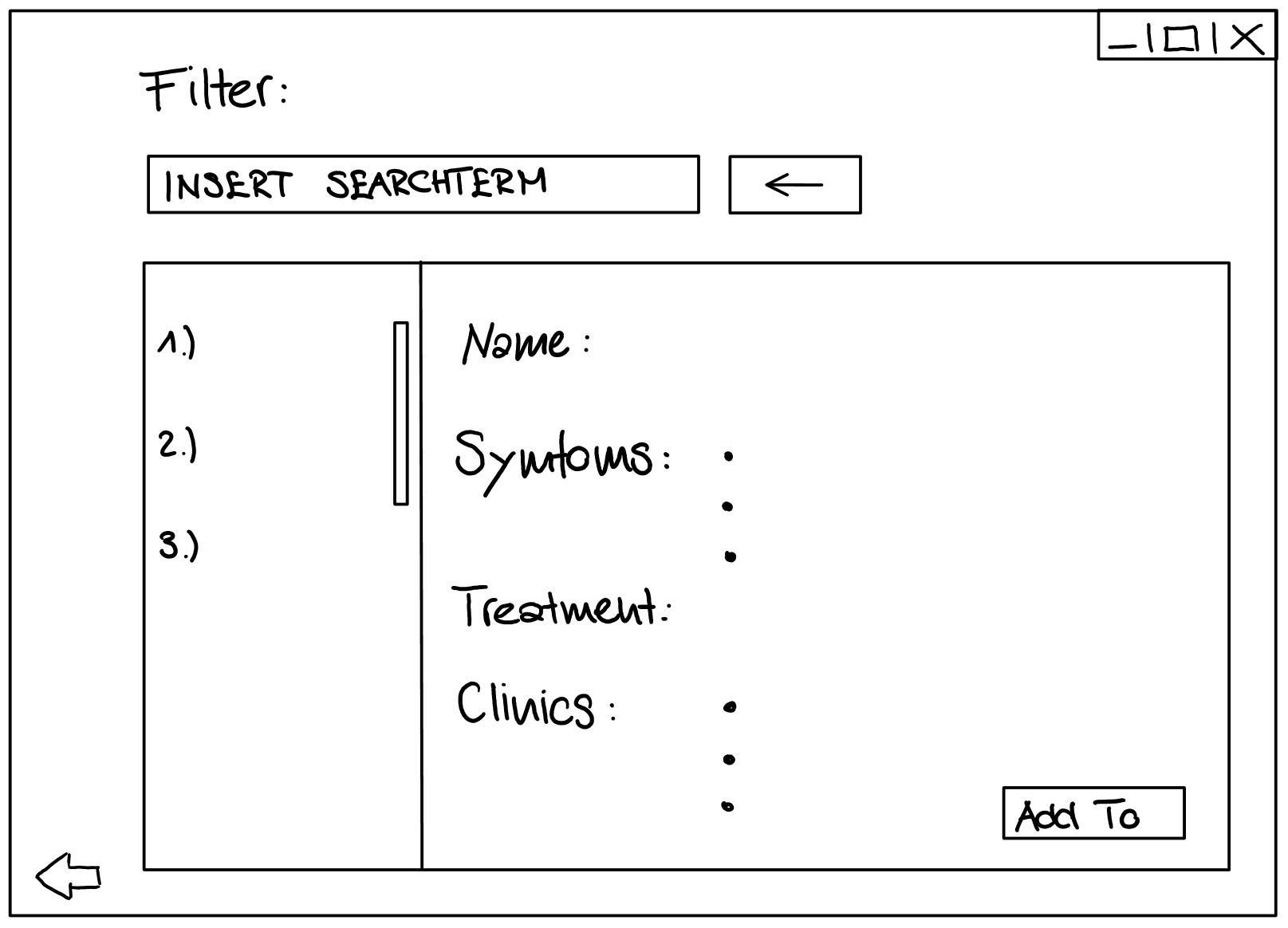


Abbildung - Use Case 7 - Sucht suchen/zuweisen

## Beschreibung Use Case 2

Im folgenden Abschnitt wird der Use Case 2 detailliert beschrieben.

|  |  |
| --- | --- |
| **Nr. und Name** | 2. Report generieren |
| **Kurzbeschreibung** | Für die wöchentliche Teamsitzung der Ärzte wird ein Report benötigt welcher die wichtigsten Fakten der letzten Wochen zusammenfasst. Das Team kann somit beurteilen ob es einen Anstieg an Terminen und somit möglichen Ressourcenmangel gibt. |
| **Beteiligte Akteure** | Ärzte, System |
| **Auslöser** | Ein Arzt benötigt eine detaillierte Zusammenfassung von verschiedenen Fakten zu Patienten, Terminen und Medikamenten. |
| **Vorbedingungen** | Die Art des benötigten Reports muss bekannt sein. |
| **Ergebnisse / Nachbedingungen** | Der Arzt erhält einen Report in einem bekannten Format und kann diesen einfach auswerten. |

Tabelle - Beschreibung Use Case 2

### Ablauf

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Wer** | **Was** |
| 2.1 | Arzt | Der Arzt klickt im Hauptmenu auf „Report“. |
| 2.2 | System | Die Reportübersicht öffnet sich und ein leeres Resultatsfeld wird angezeigt. |
| 2.3 | Arzt | Der Arzt wählt den Patient sowie die entsprechende Klinik aus und klickt auf „Generieren“. |
| 2.4 | System | Das Resultatsfeld wird durch die entsprechenden Informationen ergänzt. Der Report erscheint. |
| 2.5 | Arzt | Der Arzt kann die Resultate auswerten und begutachten. |

Tabelle - Ablauf Use Case 2

## Beschreibung Use Case 5

Im folgenden Abschnitt wird der Use Case 5 detailliert beschrieben.

|  |  |
| --- | --- |
| **Nr. und Name** | 5. Termin erfassen |
| **Kurzbeschreibung** | Infolge einer Besprechung mit einem Patienten wird ein Folgetermin vereinbart, worauf der Arzt diesen in seinem Kalender erfasst. Der Termin enthält Informationen wie Datum, Uhrzeit, Dauer und eine Beschreibung. |
| **Beteiligte Akteure** | Ärzte, System |
| **Auslöser** | Mit einem Patienten wurde ein Folgetermin vereinbart oder der Patient wird für eine Therapie in einer anderen Gesundheitsanstalt übergeben. |
| **Vorbedingungen** | Es handelt sich um keine einmalige Visite, bzw. die Behandlung dauert noch an. Das Datum, die Uhrzeit, sowie der Grund für den Termin müssen bekannt sein. |
| **Ergebnisse / Nachbedingungen** | Der Arzt trägt den Termin samt allen relevanten Informationen in der Agenda ein. Der Termin erscheint zusammen mit allen anderen Verbindlichkeiten in der Wochenübersicht der Agenda. |

Tabelle - Beschreibung Use Case 5

### Ablauf

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Wer** | **Was** |
| 5.1 | Arzt | Der Arzt klickt im Hauptmenu auf die Agenda. |
| 5.2 | System | Die Agenda öffnet sich und die Übersicht der aktuellen Arbeitswoche wird angezeigt. |
| 5.3 | Arzt | Der Arzt navigiert via Monats- oder Wochennavigation zur entsprechenden Kalenderwoche. |
| 5.4 | Arzt | Der Arzt macht einen Doppelklick auf den entsprechenden Tag. |
| 5.5 | System | Die Eingabemaske für neue Termine erscheint. Das Feld „Datum“ wird durch das System mit dem aktuellen Tag ausgefüllt. |
| 5.6 | Arzt | Der Arzt wählt aus den Dropdownfeldern den Patienten und die Klinik aus. Er gibt die Start- und Enduhrzeit ein sowie eine Beschreibung oder Notizen zum Termin. |
| 5.7 | Arzt | Wenn alle notwendigen Informationen eingegeben wurden, klickt der Arzt auf Save. |
| 5.8 | System | Der Termin wird in die Datenbank geschrieben und die Wochenansicht erscheint inkl. des neu erstellten Termins. |

Tabelle - Ablauf Use Case 5

# System Architecture

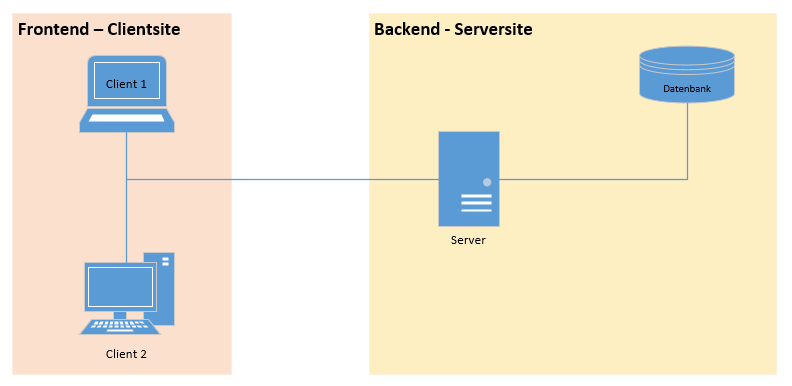


Abbildung - System Architecture

Es ist geplant, dass sowohl ein Frontend wie auch ein Backend realisiert wird. Auf dem Frontend bedienen Benutzer über ihre Clients die Applikation und nimmt Anfragen via Benutzerinput, der durch Tastatur und Maus ausgelöst wird entgegen. Die Anfragen werden an das Backend weitergeleitet, wo sie auf einem Server landen der diese weiter verarbeitet. Wenn nötig interagiert der Server mit der Applikationsdatenbank.

## Anforderungen System Architecture

Folgende Anforderungen hat die Applikation an die geplante Systemarchitektur.

### Anforderungen an Clients & Endgeräte

* Verfügen über einen modernen Browser (Firefox oder Google Chrome)
* Besitzen ein gängiges Betriebssystem wie Windows, macOS oder Linux (z.B. Ubuntu oder Fedora)
* Können eine Netzwerkverbindung zum Server herstellen

### Anforderungen an Server

* Besitzt eine aktuelle Java Installation
* Speicher liegt in ausfallsicherer Form vor (bspw. RAID1 oder RAID5)
* Webserver unterstützt Loadbalancing
* Webserver ermöglicht sichere HTTPS Verbindung

### Anforderungen an Datenbank

* Relational
* Transaktionssicher
* Daten werden zyklisch gesichert
* Zugriff kann eingeschränkt werden

# System Requirements Specification

## Funktionale Requirements

Im nachfolgenden Abschnitt werden die funktionalen System Requirements beschrieben.

### Notiz zu Patient

|  |
| --- |
| 1. Notiz erfassen |
| Der Arzt kann zu seinen Patienten eine Notiz erfassen. |

### Report

|  |
| --- |
| 2. Report generieren |
| Für eine bessere Übersicht kann der Arzt aus der Patientenübersicht einen Report eines Patienten generieren. Im Report sollten alle wichtigen Informationen wie Sucht, verschriebene Medikamenten, Termine, usw. vorhanden sein. |

### Medikamente

|  |
| --- |
| 3. Medikament suchen |
| Mit einfachen Stichwörtern soll es dem Arzt ermöglicht werden die richtigen Medikamente für einen Patienten zu suchen. |

|  |
| --- |
| 4. Medikament verschreiben |
| Der Arzt kann die Medikamente, welche dem Patienten verschrieben werden, im PMS zum Patienten hinzufügen. |

### Patientenakte

|  |
| --- |
| 5. Patient suchen |
| Der Arzt kann seine Patienten anhand verschiedener Suchkriterien wie z. B. Name, Sucht usw. suchen. |

### Agenda

|  |
| --- |
| 6. Agenda konsultieren |
| Der Arzt kann sich eine Übersicht über die Verfügbarkeit vom Personal oder seiner Termine verschaffen. |

|  |
| --- |
| 7. Termin erfassen |
| Neue vereinbarte Termine können in der Agenda eingetragen werden. |

|  |
| --- |
| 8. Termin editieren |
| Muss ein Termin verschoben werden, so kann der Arzt den entsprechenden Termin in der Agenda bearbeiten. |

|  |
| --- |
| 9. Termin löschen |
| Wird ein Termin abgesagt, kann dieser in der Agenda gelöscht werden. |

### Klinik

|  |
| --- |
| 10. Klinik suchen |
| Falls der Arzt seinen Patient nicht behandeln kann, weil er z.B. die nötige Behandlung nicht anbietet, so kann er nach einer geeigneten Klinik suchen. Das Resultat der Suche sollte die nächstgelegene Klinik sein, welche die nötige Behandlung anbietet. |

|  |
| --- |
| 11. Klinik anfragen |
| Um die Kommunikation mit anderen Kliniken zu erleichtern, kann der Arzt vom System heraus eine Mail an andere Kliniken schicken. |

### Suchtverzeichnis

|  |
| --- |
| 12. Sucht suchen |
| Der Arzt kann anhand der Symptome nach Süchten zu suchen. |

|  |
| --- |
| 13. Sucht zuweisen |
| Hat der Arzt beim Patienten eine Sucht festgestellt, so kann er dem Patienten diese Sucht im System zuweisen. |

## Nicht-funktionale Requirements

Folgende nicht-funktionale Requirements muss das Endprodukt erfüllen:

### Usability

|  |
| --- |
| Benutzeroberfläche |
| Die Benutzeroberfläche des PMS soll möglichst benutzerfreundlich und intuitiv gestaltet werden. Eine klare Menuführung inkl. Namen der Funktionen und Icons sind zentraler Bestandteil der Usabilty. |

### Performance

|  |
| --- |
| Multiuser Fähigkeit |
| Das PMS soll von mehreren Ärzten gleichzeitig verwendet werden können. Hierzu muss eine Benutzerseparierung beispielsweise in Form von Benutzerkonten realisiert werden. |

|  |
| --- |
| Performance |
| Da die verschiedenen Verzeichnisse auf eine Datenbank zugreifen, soll der Datenbankzugriff möglichst effizient und schnell erfolgen, sodass keine grösseren Wartezeiten bei den Anfragen anfallen. |

### Sicherheit

|  |
| --- |
| Autorisierter Zugriff |
| Durch die Benutzerseparierung muss gewährleistet werden, dass jeder Benutzer des Systems lediglich die für ihn bestimmten und relevanten Datensätze angezeigt bekommt. |

|  |
| --- |
| Datenschutz |
| Da das PMS sensible Daten behandelt, muss der Zugriff auf die Applikation, Datenbankzugriffe, sowie der Transfer von Daten den heutigen Sicherheitsstandards entsprechen. |

# System Models

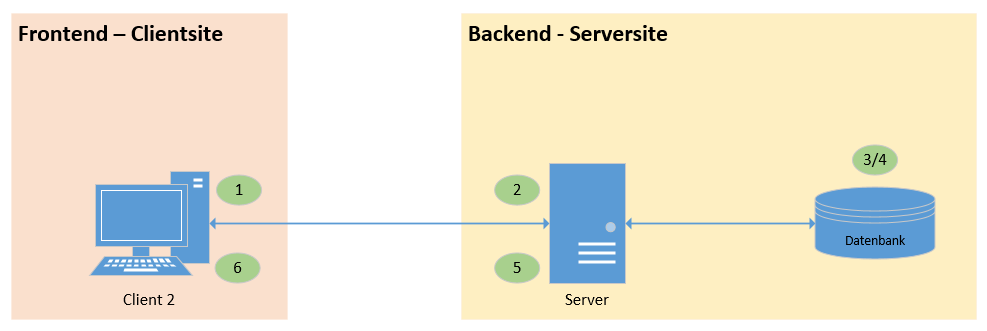


Abbildung - System Models

Auf der Grafik wird der Weg einer Anfrage von einem Endgerät dargestellt. Es soll veranschaulicht werden, was geschieht wenn der Benutzer mit seiner Oberfläche interagiert. Als konkretes Beispiel wurde der Login-Prozess gewählt, damit es auch für eine weniger technisch versierte Person verständlich ist.

|  |  |
| --- | --- |
| **Schritt / Nummer** | **Beschreibung** |
| 1 | User Request wird abgesetzt – Benutzername und Passwort werden durch Interaktion über die Benutzeroberfläche an den Server geschickt. |
| 2 | Der Server nimmt die Anfrage in Form einer Abfrage entgegen und beantragt mit einer Abfrage bei der Datenbank die Daten. |
| 3 | Die Datenbank nimmt den Request als Query entgegen, wertet diesen aus und sucht die angeforderten Daten. |
| 4 | Die Datenbank antwortet dem Server mit den angeforderten Daten. Im Fehlerfall antwortet die Datenbank mit einer entsprechenden Fehlermeldung |
| 5 | Der Server nimmt die Antwort der Datenbank entgegen und reagiert entsprechend auf die erhaltenen Daten. |
| 6 | Der Server leitet die erhaltenen und ausgewerteten Daten in einer für den Benutzer verständlichen Sprache weiter (z.B. „Login Erfolgreich“ oder „Benutzername oder Passwort falsch“. |

Tabelle - Ablauf User Request

# System Evolution

Die Anwendung soll als plattformunabhängige Webapplikation laufen. Bei der Entwicklung der Applikation werden aktuelle und offene Standards wie HTML5 und CSS3 eingesetzt. Damit soll die Kompatibilität auf der Clientseite gewährleistet werden. Serverseitig wird die Applikation mit Java entwickelt. Da die Applikation auf Windows und Linux entwickelt wird, soll die Java Applikation auf Java SE von Oracle und OpenJDK lauffähig sein. Des Weiteren soll die Applikation modular aufgebaut werden. Damit wird sichergestellt, dass bei zukünftigen Änderungswünschen des Kunden die Applikation ohne Strukturänderungen erweitert werden kann.

# Testing

Um sicherzustellen, dass die Anwendung in jeder Version möglichst fehlerfrei funktioniert, werden während der Entwicklung folgende Testverfahren durchgeführt:

* Unit-Tests  
  Programmierte Tests die laufend ausgeführt werden und so Auskunft darüber geben, ob ein Fehler während der Entwicklung unbemerkt geblieben ist. Sie geben Auskunft über die Korrektheit der Funktionalitäten der einzelnen Klassen und Methoden.
* Manuelle Tests  
  Während der Realisierung testet jeder Entwickler die Logik und das GUI. Zusätzlich wird von einem zweiten Entwickler ein manueller Test durchgeführt um die Fehlerquote weiter zu verringern („4 Augenprinzip“).
* Integrationstest  
  Nach jedem Sprint wird die aktuelle Version der Anwendung auf das Integrationssystem geladen, damit der Kunde die Anwendung zeitnah testen kann und direkt ein Feedback geben kann.
* Lasttests  
  Anhand simulierter Zugriffe soll die Belastung des Systems bei der Benutzung durch mehrere Benutzer bzw. im täglichen Betrieb simuliert und getestet werden.