

12/15

CS2 Task 2 Requirements

Team Black

Ramon Luginbühl, Luca Beyeler, Mario Schläppi, Robin Füglistner

Inhaltsverzeichnis

| | | |
|-----|---|----|
| 1 | Preface..... | 4 |
| 2 | Introduction..... | 5 |
| 3 | Glossary | 6 |
| 3.1 | Glossary zum Datenbank Entwurf..... | 7 |
| 4 | User and System Requirements Specification..... | 8 |
| 4.1 | System Boundary..... | 8 |
| 4.2 | Use Case 1 / Erstellung private Event..... | 9 |
| 4.3 | Use Case 2 / Teilnahme an public Event | 10 |
| 4.4 | Use Case 3 / Eventtemplate suchen..... | 11 |
| 4.5 | Use Case 4 / Versenden und akzeptieren einer Freundschaftsanfrage | 12 |
| 4.6 | Use Case 5 / Erstellung eines public Events | 12 |
| 5 | Functional Requirements | 13 |
| 5.1 | Account erstellen und benutzen | 13 |
| 5.2 | Patient suchen/hinzufügen/Info | 13 |
| 5.3 | Events durchsuchen | 13 |
| 5.4 | Event erstellen..... | 13 |
| 5.5 | Event ändern und bewerten | 13 |
| 6 | Non-Functional Requirements | 14 |
| 6.1 | Anonymität und Datenschutz von Nutzern..... | 14 |
| 6.2 | DB kann nicht öffentlich durchsucht werden..... | 14 |
| 6.3 | Spamprävention | 14 |
| 6.4 | Plattformunabhängigkeit | 14 |
| 6.5 | Interface | 14 |
| 7 | System Architecture | 15 |
| 7.1 | Modules..... | 15 |
| 7.2 | Architecture..... | 15 |
| 7.3 | System Models | 16 |
| 8 | System Evolution | 17 |
| 8.1 | Lokale Version | 17 |
| 8.2 | Web Version | 17 |
| 8.3 | System Generell..... | 17 |
| 9 | Testing | 17 |
| 9.1 | Unit Tests (White Box)..... | 17 |
| 9.2 | Integration Tests (White Box) | 17 |
| 9.3 | Application Tests (Black Box)..... | 17 |

| | | |
|------|-------------------|----|
| 9.4 | Story Tests | 17 |
| 10 | Appendices | 18 |
| 11 | Index | 18 |
| 11.1 | Pictures..... | 18 |
| 11.2 | Tables..... | 18 |

1 Preface

Dieses Dokument dient zur Definition aller Requirements an die Event Webapp für Burnout Patienten. Das Dokument wird von folgenden Personen verwendet:

- Patienten
- Relatives
- Entwickler der Software (Team black)
- Auftraggeber PMS

| Version | Rationale | Changes |
|-------------|---|---|
| V0.1 | <ul style="list-style-type: none">- Erste Gedanken- Einteilung der Abschnitte- Ideensammlung | <ul style="list-style-type: none">- Kapitel gemäss Task 2 erstellt- Use Cases definiert- Erste functional / non-functional Requirements wurden definiert bzw. aus Task 1 übernommen- Use Case Detaillierung wurde begonnen |
| V0.2 | <ul style="list-style-type: none">- Fixierung der Use Cases- Erneute Besprechung der Ideensammlung- Besprechung der möglichen System Requirements | <ul style="list-style-type: none">- Preface und Introduction wurden erstellt- Finalisierung der Use Cases- Erstellung des Use Case Diagramms- Definierung der ersten System Requirements |
| V0.3 | <ul style="list-style-type: none">- Besprechung der System Architektur- System Requirements Erweiterung | <ul style="list-style-type: none">- Finalisierung der System Requirements- System Models wurden begonnen- System Evolution wurde besprochen und erste Gedanken zum Testing festgehalten |
| V0.4 | <ul style="list-style-type: none">- Iteration der einzelnen Kapitel- Aufgetretene Fragen besprochen | <ul style="list-style-type: none">- Ergänzungen zu diversen Kapiteln- System Evolution und Testing wurde finalisiert- System Models vervollständigt- Appendices wurden erstellt |
| V0.5 | <ul style="list-style-type: none">- Letzte Iteration der Themen | <ul style="list-style-type: none">- Finalisierung aller offenen Themen- Indexes erstellt |
| V1.0 | <ul style="list-style-type: none">- Abschluss des Dokuments- Review der Themen- Glossary erstellen | <ul style="list-style-type: none">- Glossary wurde erstellt und finalisiert- Rechtschreibung geprüft / Formatierungen- Finale Version wurde erstellt- Version History vervollständigt |

Tabelle 1 History

2 Introduction

Das Patienten Management System benötigt eine externe Webapp, um Burnout Patienten zu unterstützen, die Depersonalisierung während der Behandlung zu verringern. Es gibt momentan keine unterstützende Lösung für dieses Anliegen. Die Relatives der Burnout Patienten bekommen mit dieser Webapp die Möglichkeit, sich aktiv an diesem Teil der Behandlung zu beteiligen. Zudem sollen die Events als Informationsaustausch zwischen den Relatives und den Patienten sowie in der Beziehung Patient zu Patient / Relative zu Relative dienen. So soll das Verständnis des Umfelds gefördert werden sowie die Patienten motiviert werden sich aktiv am Leben zu beteiligen.

Funktionalitäten der Webapp:

- Erstellung von privaten Events durch einen Relative mit einem Patienten
- Hinzufügen von Patienten durch einen Relative via Freundschaftsanfrage
- Teilnahme und Erstellung von public Events mit mehreren Relatives / Patienten
- Austausch zwischen den Personen via Chat Funktion (basierend auf den Events, kein direkter Chat)
- Erstellung von Events via Templates → Templates werden aus vergangenen abgeschlossenen Events generiert
- Filterung der Templates
- Verwaltung der Events
- Rating der Events bei Abschluss

Es wird keine direkte Schnittstelle zum PMS erstellt. Die behandelnden Ärzte können die Patienten befragen, um die Resultate der Events in die Behandlung einfließen zu lassen. Die Webapp soll via Webbrowser verwendet werden können.

Die regionale Gesundheitsbehörde will den Patienten eine flexiblere Unterstützung bei der Bewältigung des Alltags bieten und eine neue Behandlungsart einführen. Durch diese Webapp soll dies möglichst benutzerfreundlich ermöglicht werden. Durch die Planung der Events sollen die Burnout Patienten bei der Wiedereingliederung unterstützt werden und auch die behandelnden Ärzte entlastet werden.

3 Glossary

| Wort | Beschreibung |
|------------------------------------|--|
| APP | Applikation / Programm |
| Black Box | Methode eines Software Tests (Quellcode ist verborgen) |
| Client | Programm und Anwender, welche die Dienste des Servers in Anspruch nehmen |
| Databases | Datenbanken |
| DB | Datenbank |
| Event | Unternehmung |
| Functional Requirements | Funktionelle Anforderungen (z.B. Account erstellen) |
| Interface | Schnittstelle |
| Mobile-App | Applikation für mobile Geräte |
| Non-Functional Requirements | Nicht funktionelle Anforderungen (z.B. Datenschutz) |
| PMS | Patienten Management System à <u>Verwaltung</u> der Patientendaten, Behandlung, Medikation, etc. |
| Private Event | Privates Event (private Unternehmung mit Patient) |
| Public Event | Öffentliches Event (öffentliche Unternehmung mit Patient) |
| Relative | Bekannter (hier Bekannter eines Burnout Patienten) / <i>Angehörigen</i> |
| Server | Rechner welcher ein Dienst (hier: Patient Management System/Datenbank) via Netzwerk bereitstellt |
| System | Applikation und Logik auf dem Server |
| Template | Vorlage (hier Vorlage eines Events) |
| Webapp | Webanwendung, wird via Webbrowser bedient |
| White Box | Methode eines Software Tests (Blick auf Quellcode) |

Tabelle 2 Glossary allgemein

3.1 Glossary zum Datenbank Entwurf

| Wort | Beschreibung |
|------------------------|--------------------------------------|
| Bool | Datentyp (Binär) |
| ID | Identifikator |
| Integer | Datentyp (Nummer) |
| MaxParticipants | Maximale Anzahl Teilnehmer |
| Rating | Bewertung eines Events |
| Rel / Relation | Beziehung zwischen Datenbanktabellen |
| State | Status (Offen/Erledigt/Abgesagt) |
| Tag | Stichwort, um Events zu filtern |
| VarChar | Datentyp (Text) |

Tabelle 3 Glossary zu DB Entwurf



4 User and System Requirements Specification

4.1 System Boundary

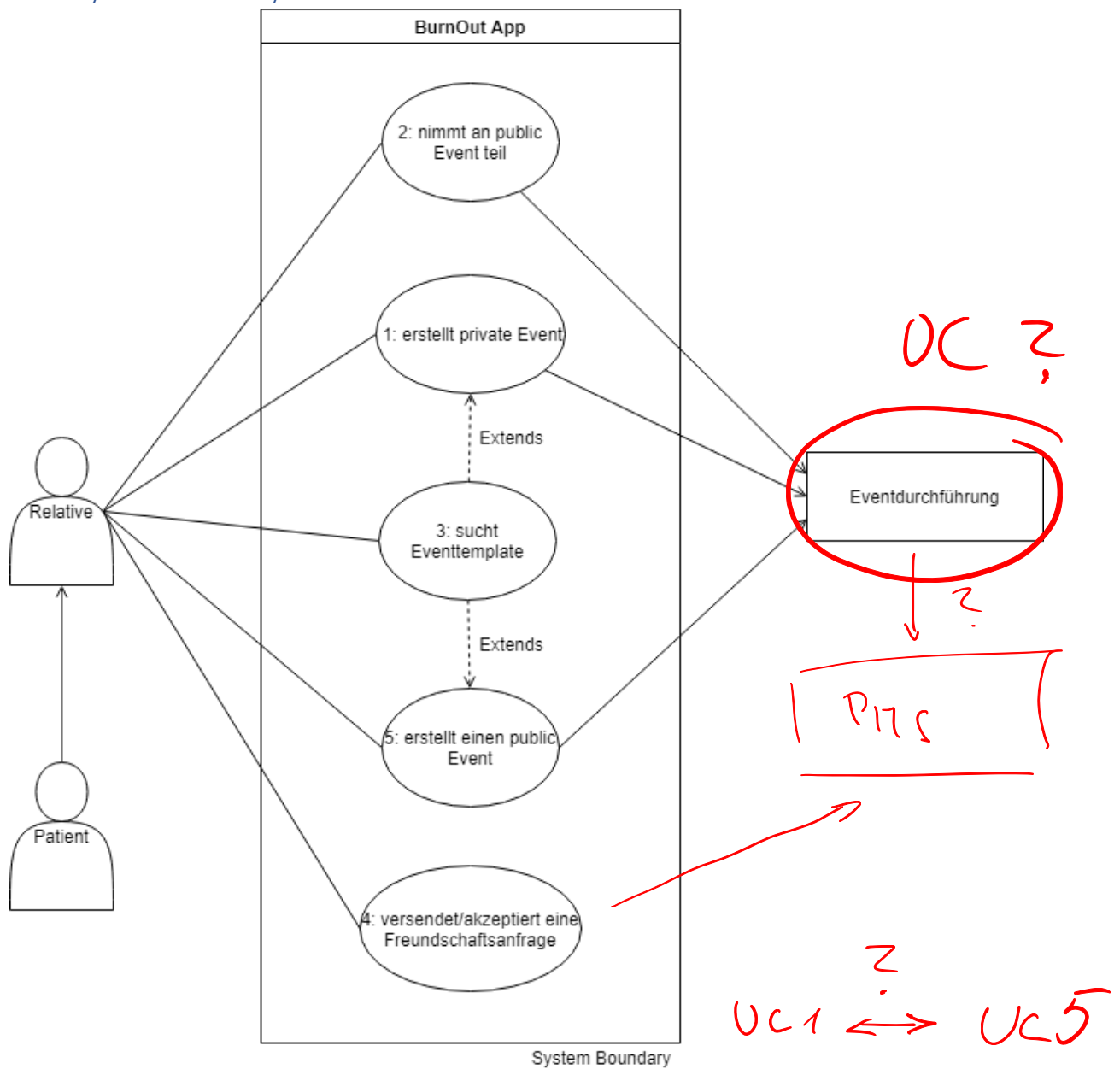


Abbildung 1 System Boundary

- Actor Beschreibung
- Abgrenzung

4.2 Use Case 1 / Erstellung private Event

| | |
|-----------------------------|--|
| Nr. und Name: | Use Case 1 / Erstellung private Event |
| Szenario: | Ein privates Event soll vom Relative erstellt werden und gemeinsam mit dem Patienten durchgeführt werden. |
| Kurzbeschreibung: | Ein Relative erstellt in der Event Webapp ein privates Event, fügt einen Patienten dem Event hinzu, führt das Event durch und schliesst dies ab. |
| Beteiligt Akteure: | Relative, Patient und System |
| Auslöser / Vorbedingung: | Relative erstellt das private Event, Relative kennt eine Aktivität |
| Ergebnisse / Nachbedingung: | Event wurde erstellt, durchgeführt und abgeschlossen |

Tabelle 4 Use Case 1 Header

Ablauf:

| Nr. | Wer | Was |
|------|--------------------|---|
| 1.0 | Relative | Navigiert via Startseite auf die Maske zur Erstellung eines neuen Events |
| 1.1 | System | Legt ein neues Event Objekt an |
| 1.2 | Relative | Füllt Formularfelder aus, wählt die Option „private“ aus und klickt auf die Patientenauswahl |
| 1.3 | System | Checkt welche Patienten bereits eine Beziehung mit dem Relative haben und stellt diese zur Auswahl zur Verfügung. Zusätzlich wird die Patientensuche geladen. |
| 1.4 | Relative | Patient wird dem Event hinzugefügt, ggf. prüft Relative die Aktivitäten des gewünschten Patienten |
| 1.5 | System | Patient wird dem Event Objekt zugewiesen und angezeigt |
| 1.6 | Relative | Erstellt das vollständig ausgefüllte Event |
| 1.7 | System | Ausnahme: Nicht alle Angaben ausgefüllt |
| 1.8 | System | Weist das Event dem Relative sowie dem Patienten zu, sodass dies unter „My Events“ ersichtlich ist |
| 1.9 | Relative / Patient | Event wird physisch durchgeführt, Foto der Durchführung wird aufgenommen <i>↳ System startet Kamera</i> |
| 1.10 | Relative / Patient | Unter „My Events“ wird das durchgeführte Event auf erledigt gesetzt |
| 1.11 | System | Öffnet die Eventabschluss Maske |
| 1.12 | Relative / Patient | Lädt das Foto in dem Form hoch (Formate sind vorgegeben keine falschen Formate möglich), vergibt einen Score und schliesst das Event ab |
| 1.13 | System | Ausnahme: Nicht alle Angaben sind ausgefüllt |
| 1.14 | System | Event Objekt wird als abgeschlossen markiert, dem Relative wird der entsprechende Fortschritt zugewiesen |
| 1.15 | System | Event wird als Template anonymisiert in die Event Datenbank geschrieben und dem Event Template wird der vergebene Score zugewiesen |
| 1.16 | System | Der neue Fortschritt wird dem Relative angezeigt |

Tabelle 5 User Case 1 Schritte

Ausnahmen, Varianten:

| Nr. | Wer | Was |
|--------|----------|---|
| 1.7 | | Nicht alle Angaben ausgefüllt |
| 1.7.1 | System | Fehlermeldung mit Angabe der nicht korrekten / vollständigen Daten, Abbruch des Vorganges |
| 1.7.2 | Relative | Korrektur / Ergänzung der Daten |
| 1.7.3 | Relative | Löst die Erstellung des Events nochmals aus |
| 1.13 | | Nicht alle Angaben ausgefüllt |
| 1.13.1 | System | Fehlermeldung mit Angabe der nicht korrekten / vollständigen Daten, Abbruch des Vorganges |
| 1.13.2 | Relative | Korrektur / Ergänzung der Daten |
| 1.13.3 | Relative | Löst den Abschluss des Events nochmals aus |

*Tabelle 6 Use Case 1 Ausnahmen***4.3 Use Case 2 / Teilnahme an public Event**

| | |
|-----------------------------|--|
| Nr. und Name: | Use Case 2 / Teilnahme public Event |
| Szenario: | Der Relative nimmt mit dem Patienten an einem public Event teil, an dem mehrere Personen teilnehmen. |
| Kurzbeschreibung: | Der Relative sucht in der „Open Public Events“ Sektion nach einem passenden Event und trägt sich und seinen Patienten für das public Event ein. Die Teilnehmer besprechen die Details und Durchführung via Chat. Die Event Durchführung und der Abschluss ist analog Use Case 1. |
| Beteiligt Akteure: | Relative, Patient und System |
| Auslöser / Vorbedingung: | Relative sucht nach dem passenden public Event und schreibt sich und den entsprechenden Patienten ein. Ein public Event muss zur Verfügung stehen. |
| Ergebnisse / Nachbedingung: | Teilnahme wurde registriert, Event durchgeführt und abgeschlossen |

Tabelle 7 Use Case 2 Header

4.4 Use Case 3 / Eventtemplate suchen

| | |
|-----------------------------|---|
| Nr. und Name: | Use Case 3 / Eventtemplate suchen |
| Szenario: | Der Relative möchte sich von Ideen von anderen Relatives inspirieren lassen und durchsucht die bewerteten Event-Templates. |
| Kurzbeschreibung: | Der Relative sucht in den Event Templates nach Ideen. Falls ihm eine Idee gefällt, kann er direkt das Template als Maske für die Erstellung eines Events verwenden. |
| Beteiligt Akteure: | Relative, Patient und System |
| Auslöser / Vorbedingung: | Relative möchte etwas mit seinem Patienten unternehmen, hat aber gerade nicht die passende Idee. |
| Ergebnisse / Nachbedingung: | Passendes Event wurde gefunden und durchgeführt. |

Tabelle 8 Use Case 3 Header

Ablauf:

| Nr. | Wer | Was |
|-----|----------|--|
| 1.0 | Relative | Navigiert via Startseite auf die Maske Event Templates <i>-> System zeigt Suchmaske</i> |
| 1.1 | Relative | Füllt die Suchmaske mit gewünschten Begriffen und/oder sortiert Events nach Kriterien |
| 1.2 | System | Filtert / sortiert Events entsprechend und zeigt diese an |
| 1.3 | Relative | Wählt passendes Event als Template |
| 1.4 | System | Erstellt ein neues Event Objekt und befüllt die Maske mit der Beschreibung aus dem Template. |
| 1.5 | Relative | Entscheidet ob der Event private oder public ist und klickt auf die Patientenauswahl |
| 1.6 | | Weitere Schritte analog Use Case 1 ab 1.3 ✓ |

Tabelle 9 Use Case 3 Schritte

Ausnahmen, Varianten:

| Nr. | Wer | Was |
|-----|----------|--|
| 1.3 | Relative | Bricht die Suche ab, ohne ein Event zu erstellen |

Tabelle 10 Use Case 3 Ausnahmen

4.5 Use Case 4 / Versenden und akzeptieren einer Freundschaftsanfrage

| | |
|-----------------------------|--|
| Nr. und Name: | Use Case 4 / Versenden und Akzeptieren einer Freundschaftsanfrage |
| Szenario: | Ein Relative möchte ein Event planen mit einem Patienten, zu welchem noch keine Beziehung besteht bzw. noch keine Freundschaftsanfrage versendet wurde oder der Relative sucht einen neuen Patient. |
| Kurzbeschreibung: | In der Event Erstellung oder direkt auf der Startseite unter « <u>Search Patient</u> » sucht der Relative nach einem neuen Patienten und fügt diesen via Freundschaftsanfrage hinzu. Der Patient akzeptiert die Freundschaftsanfrage bzw. lehnt diese ab. Der Patient steht anschliessend dem Relative zur Verfügung bei der Erstellung von Events und anderen Funktionen. |
| Beteiligt Akteure: | Relative, Patient und System |
| Auslöser / Vorbedingung: | Der Relative versendet die Freundschaftsanfrage. Es muss einen Patienten geben, mit dem der Relative noch nicht befreundet ist. |
| Ergebnisse / Nachbedingung: | Eine Beziehung zwischen Relative und Patient wurde hergestellt, wenn die Freundschaftsanfrage akzeptiert wurde. |

→ PMS

Tabelle 11 Use Case 4 Header

4.6 Use Case 5 / Erstellung eines public Events

| | |
|-----------------------------|--|
| Nr. und Name: | Use Case 5 / Erstellung eines public Events |
| Szenario: | Ein public Event soll vom Relative erstellt werden und gemeinsam mit dem Patienten und anderen Teilnehmer durchgeführt werden. |
| Kurzbeschreibung: | Ein Relative erstellt in der Event Webapp ein public Event, fügt einen Patienten dem Event hinzu, wartet auf weitere Teilnehmer, bespricht mit den Teilnehmern die Details des Events via Chat, führt das Event durch und schliesst dies ab. |
| Beteiligt Akteure: | Relative, Patient und System |
| Auslöser / Vorbedingung: | Relative erstellt das public Event, es müssen sich entsprechend viele andere Relatives gemäss Use Case 2 eingeschrieben haben bzw. teilnehmen |
| Ergebnisse / Nachbedingung: | Event wurde erstellt, durchgeführt und abgeschlossen |

Tabelle 12 Use Case 5 Header

5 Functional Requirements

→ Abgrenzung/Bezug
Use Case Requirements?

5.1 Account erstellen und benutzen

E-Mail als Benutzername und Passwort wird angefragt und anschliessend wird Account erstellt. Mit dem Account können meine Events eingesehen und die darin enthaltene Chatfunktion genutzt werden.

5.2 Patient suchen/hinzufügen/Info

DB wird nach Patientennamen durchsucht. Patient kann zu meinen Patienten durch eine Freundschaftsanfrage hinzugefügt werden. Infos zu den Ausprägungen der Krankheit, Handicaps und Wünsche der Patienten. Diese Infos werden vom Medizinischen Personal und dem Patienten erstellt.

5.3 Events durchsuchen

Öffentliche Events nach Belieben durchsuchen und Filtern. Sich gegebenenfalls mit seinem Patienten einschreiben.

5.4 Event erstellen

Eigenes privates oder öffentliches Event mit eigener Idee oder aus einem Template erstellen.

5.5 Event ändern und bewerten

Von mir erstellte Events abändern, raten und abschliessen können.

6 Non-Functional Requirements

6.1 Anonymität und Datenschutz von Nutzern

Relatives können erst nach bestätigter Freundschaftsanfrage mit Patienten in Kontakt treten. Die Chatfunktion existiert nur pro Event und wird in den Event Details angehängt. Der Chat ist jeweils für die entsprechenden Eventteilnehmer sichtbar.

Die E-Mail wird nie angezeigt und dient nur zur Autorisierung. Nur der Name der Teilnehmer ist jeweils sichtbar. Weitere Kontaktmöglichkeiten müssen eigenständig über den Chat ausgetauscht werden.

6.2 DB kann nicht öffentlich durchsucht werden

Suchmaschinen erreichen keine Nutzerspezifischen Informationen der Webapp.

6.3 Spamprävention

Nur eingeschränkte Eventkreationen pro Relative möglich

6.4 Plattformunabhängigkeit

Die App muss Plattform unabhängig genutzt werden können

6.5 Interface

Durchgeführte Events werden mit Infos an PMS weitergeleitet.

?

↳ genauer + quantifizieren

7 System Architecture

7.1 Modules

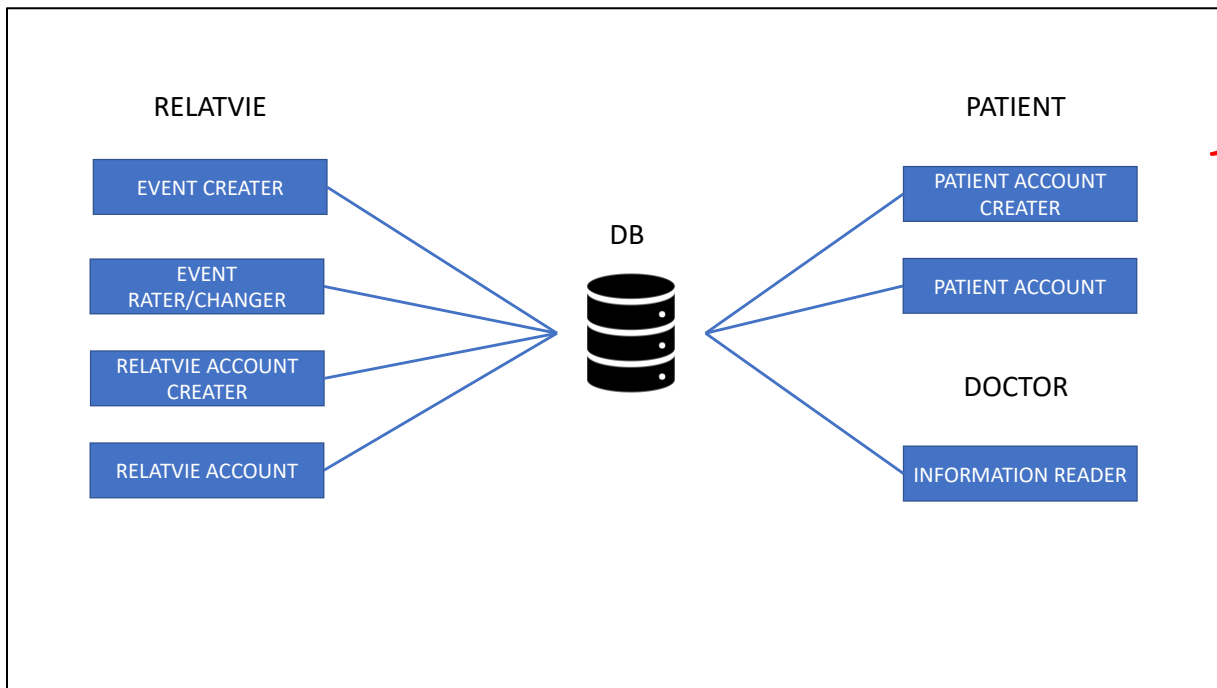


Abbildung 2 Modules

7.2 Architecture

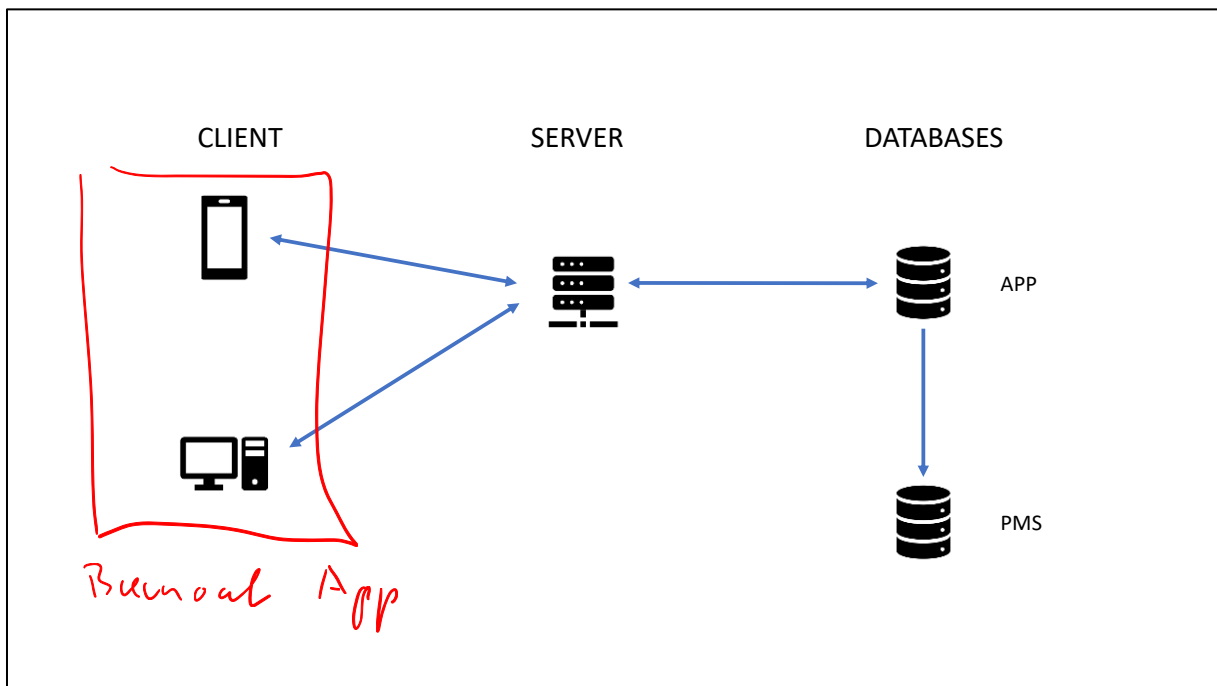


Abbildung 3 Architecture

7.3 System Models

Event DB Entwurf

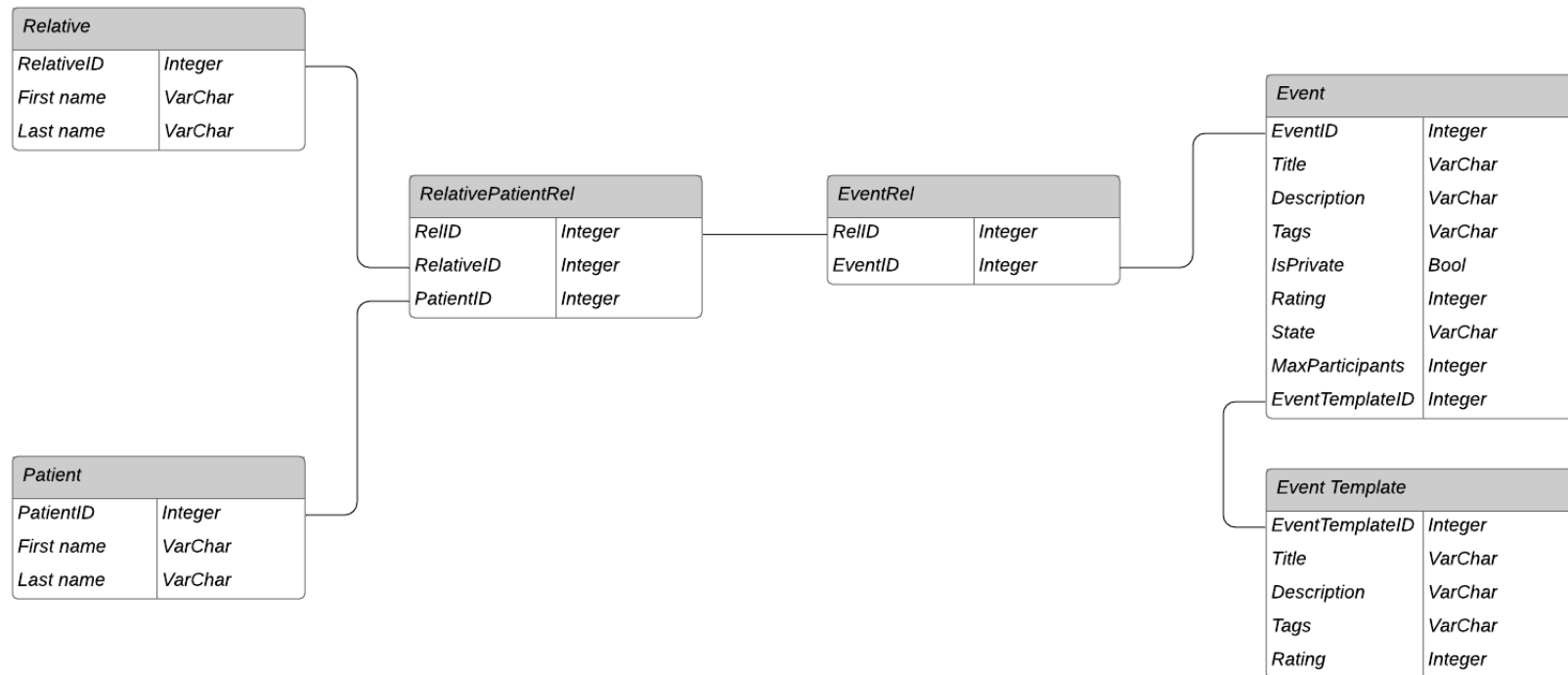


Abbildung 4 Event DB Entwurf

8 System Evolution

8.1 Lokale Version

Um die Entwicklung einfacher zu machen wird die Applikation erst lokal aufgebaut.

8.2 Web Version

Sobald eine stabile Version der Software bereitsteht, wird die Seite mit einem Webserver im Hintergrund online gestellt. Der Zugriff auf die Webseite kann von jeglichen Geräten erfolgen (Computer, Handy, Tablet, etc.). Es wird jedoch keine App-Version explizit für Handys implementiert.

8.3 System Generell

Für weiterführende Funktionen gibt es viele Möglichkeiten, wie beispielsweise:

- Ideenpool
- Mobile App-Version

9 Testing

9.1 Unit Tests (White Box)

Diese Tests haben das Ziel einzelne Komponenten, Klassen oder Methoden auf ihre Funktionalität und Korrektheit zu prüfen. Diese werden nach jedem Build und Check-In automatisch ausgeführt. Eine Erfolgsquote von 100% ist dabei Voraussetzung für die Fortsetzung.

9.2 Integration Tests (White Box)

Diese Tests haben das Ziel die Funktionalität der verschiedenen Komponenten untereinander zu verifizieren. Dies geschieht ebenfalls automatisch und im Falle eines Fehlers muss dieser anschliessend erst eliminiert werden.

9.3 Application Tests (Black Box)

Diese Tests haben das Ziel das gesamte System zu prüfen. Hauptsächlich soll hier automatisch getestet werden ob Applikation und Datenbank korrekte Antworten auf spezifische Anfragen liefern.

9.4 Story Tests

Die abgeschlossenen Use Cases sollen von einem Tester manuell geprüft werden sobald diese umgesetzt wurden. Hier geschieht alles manuell.



10 Appendices

Serverseitig wird die Software inklusive der Datenbank in ein bestehendes IT-System integriert. Im Falle einer ungenügenden Performance muss die bestehende Infrastruktur erweitert werden können. Auf der Seite des Clients wird darauf geachtet, dass möglichst jede gängige Hardware (Tablett, Mobiltelefon oder PC/Mac) unterstützt wird.

11 Index

11.1 Pictures

| Bild | Seite |
|-------------------------------------|-------|
| Abbildung 1 System Boundary | 8 |
| Abbildung 2 Modules | 15 |
| Abbildung 3 Architecture | 15 |
| Abbildung 4 Event DB Entwurf | 16 |

11.2 Tables

| Tabelle | Seite |
|--|-------|
| <i>Tabelle 1 History</i> | 4 |
| <i>Tabelle 2 Glossary allgemein</i> | 6 |
| <i>Tabelle 3 Glossary zu DB Entwurf</i> | 7 |
| <i>Tabelle 4 Use Case 1 Header</i> | 9 |
| <i>Tabelle 5 User Case 1 Schritte</i> | 9 |
| <i>Tabelle 6 Use Case 1 Ausnahmen</i> | 10 |
| <i>Tabelle 7 Use Case 2 Header</i> | 10 |
| <i>Tabelle 8 Use Case 3 Header</i> | 11 |
| <i>Tabelle 9 Use Case 3 Schritte</i> | 11 |
| <i>Tabelle 10 Use Case 3 Ausnahmen</i> | 11 |
| <i>Tabelle 11 Use Case 4 Header</i> | 12 |
| <i>Tabelle 12 Use Case 5 Header</i> | 12 |