**Analyse & Konzeption**

**Synth UI**

Verfasser: Schmidt, Julian Alexander

Version: V1.0

Status: Öffentlich

Github Repository: <https://github.com/Julian-AT/synth-ui>

Abteilung für Informatik – Software-Engineering, 5AHIF

Schuljahr 2024/25

0. Inhaltsverzeichnis

Inhalt

[0. Inhaltsverzeichnis 2](#_Toc180533634)

[1. Einleitung 3](#_Toc180533635)

[1.1 Zweck und Gültigkeit des Dokuments 3](#_Toc180533636)

[1.2 Projektübersicht 3](#_Toc180533637)

[1.3 Ziele und Erwartungen 3](#_Toc180533638)

[2. Projektgegenstand 5](#_Toc180533639)

[2.1 Projektumfang & Abgrenzungen 5](#_Toc180533640)

[3. Analyse und Konzeption 6](#_Toc180533641)

[3.1 IST-Analyse 6](#_Toc180533642)

[3.1.1 Herausforderung und Problemstellung 6](#_Toc180533643)

[3.1.2 Stakeholder-Analyse 7](#_Toc180533644)

[3.1.3 Nutzerprofil und Personas 8](#_Toc180533645)

[3.1.4 Lösungsansatz PoC 10](#_Toc180533646)

[3.2 SOLL-Prozess der KI-Unterstützung 11](#_Toc180533647)

[3.2.1 Datenansicht 11](#_Toc180533648)

[3.2.2 Nutzersicht 12](#_Toc180533649)

[3.2.3 Systemarchitektur 12](#_Toc180533650)

[3.3 Technische Verifikation des KI-Einsatzes 14](#_Toc180533651)

[3.3.1 Datengrundlage und -qualität 14](#_Toc180533652)

[3.3.2 Einsatzbereich und Limitationen 14](#_Toc180533653)

[3.4 Erfolgskriterien und KPIs 15](#_Toc180533654)

[3.4.1 Quantitative Merkmale 15](#_Toc180533655)

[3.4.2 Qualitative Merkmale 16](#_Toc180533656)

[3.4.3 Testszenarien und Benchmarks 16](#_Toc180533657)

[4. Technische Spezifikation 17](#_Toc180533658)

[4.1 Softwarearchitektur & Datenmodell 17](#_Toc180533659)

[5. Implementierung 18](#_Toc180533660)

[5.1 Code Standards 18](#_Toc180533661)

[5.2 CI/CD & DevOps 18](#_Toc180533662)

1. Einleitung

## 1.1 Zweck und Gültigkeit des Dokuments

Das vorliegende Dokument dient als umfassende und autoritative Quelle für alle Informationen zum Projekt Synth UI. Es beschreibt detailliert die Funktionsweise, die technischen Grundlagen sowie die Einsatzmöglichkeiten der Anwendung. Ziel ist es, eine klare und strukturierte Übersicht zu bieten, die sowohl Entwicklern als auch anderen Stakeholdern einen tiefen Einblick in das Projekt ermöglicht.

Die Gültigkeit dieses Dokuments bezieht sich auf alle Versionen von Synth UI ab Version 1.0 und bildet die Grundlage für zukünftige Weiterentwicklungen und den Support. Mit jeder neuen Version wird das Dokument entsprechend angepasst und erweitert, um sicherzustellen, dass es stets aktuelle und relevante Informationen enthält.

## 1.2 Projektübersicht

Synth UI ist eine AI-basierte SaaS-Anwendung, die Entwicklern ermöglicht, React-Komponenten durch natürliche Sprachbefehle zu generieren. Mithilfe fortschrittlicher KI-Technologien wandelt die Anwendung textuelle Anweisungen in fertige React-Komponenten um, was den Entwicklungsprozess erheblich beschleunigt und vereinfacht. Das System richtet sich an Entwickler, die wiederkehrende UI-Elemente effizienter erstellen möchten, sowie an Teams, die die Kommunikation zwischen Design und Entwicklung optimieren wollen.

## 1.3 Ziele und Erwartungen

Synth UI verfolgt das Ziel, die Entwicklung von Benutzeroberflächen durch den Einsatz von KI erheblich zu vereinfachen und zu beschleunigen. Im Zentrum steht die automatische Generierung von React-Komponenten auf Basis natürlicher Sprachbefehle. Dadurch sollen sowohl Effizienz als auch die Qualität in der Frontend-Entwicklung gesteigert werden.

**Primäre Ziele:**

* **Automatisierung der UI-Entwicklung:** Durch die Konvertierung von natürlicher Sprache in Code sollen wiederkehrende Aufgaben, wie das Erstellen von UI-Komponenten, automatisiert werden. Dies spart Entwicklern Zeit und reduziert menschliche Fehler.
* **Reduktion der technischen Einstiegshürde/Lernkurve:** Durch die Nutzung von Synth UI sollen auch Personen ohne tiefe Programmierkenntnisse in der Lage sein, funktionsfähige React-Komponenten zu erstellen.
* **Steigerung der Produktivität:** Teams können mithilfe der Plattform schneller Ergebnisse erzielen, indem sie Routineaufgaben durch KI unterstützen lassen, sodass der Fokus stärker auf kreativen und komplexen Aufgaben liegt.
* **Erhöhte Wiederverwendbarkeit:** Die durch Synth UI erstellten Komponenten sollen modular und wiederverwendbar sein, was zu einem effizienteren Entwicklungszyklus beiträgt.

**Erwartungen an Synth UI:**

* **Qualität der generierten Komponenten:** Synth UI soll qualitativ hochwertige, wartbare und standardkonforme React-Komponenten erzeugen, die direkt in bestehende Projekte integriert werden können.
* **Nahtlose Integration in den Entwicklungsprozess:** Synth UI soll sich problemlos in bestehende Entwicklungsumgebungen und Workflows integrieren lassen, um eine maximale Produktivität zu gewährleisten.
* **Anpassbarkeit der generierten Komponenten:** Nutzer sollen die Möglichkeit haben, die erstellten UI-Elemente nach Bedarf anzupassen und zu verfeinern, um spezifischen Projektanforderungen gerecht zu werden.
* **Kontinuierliche Verbesserung:** Das System soll durch kontinuierliches Training des zugrundeliegenden Modells und durch die Integration von Benutzerfeedback verbessert werden, um sich neuen Anforderungen anzupassen.

2. Projektgegenstand

## 2.1 Projektumfang & Abgrenzungen

Synth UI umfasst in der Version V1.0 folgende Aspekte:

* **Text-to-Code-Generierung:** Automatische Umwandlung von natürlichen Sprachbefehlen in React-Komponenten.
* **Anpassbare Komponenten:** Möglichkeit zur flexiblen Anpassung der generierten UI-Elemente.
* **Nutzung verschiedener LLMs:** Synth UI ist so konzipiert, dass es mit verschiedenen Large Language Models (LLMs) kompatibel ist.
* **Komponenten-Datenbank:** Zugriff auf eine Wissensdatenbank, die 5 UI-Libraries mit über 100 wiederverwendbaren Komponenten enthält.
* **Skalierbare SaaS-Architektur:** Bereitstellung einer cloudbasierten und skalierbaren Lösung, die mit den Anforderungen wachsender Teams und Projekte mitwächst.
* **Benutzerfreundliche Oberfläche:** Fokus auf eine intuitive Benutzeroberfläche, die sowohl für Entwickler als auch Nicht-Entwickler geeignet ist.

Abgrenzungen von Synth UI zur präzisen Definition des funktionalen und technologischen Rahmens des Projekts:

* **Einsatz von LLMs:** Das System unterstützt eine Vielzahl von Large Language Models (LLMs), ist jedoch nicht auf spezifische oder proprietäre Modelle beschränkt.
* **Automatisierung:** Während Synth UI die Generierung von Komponenten erleichtert, liegt die Verantwortung für die finale Implementierung und Anpassung der generierten UI-Elemente beim Entwickler.
* **Technologischer Fokus:** Die Anwendung ist speziell auf die Erstellung von React-Komponenten ausgelegt und bietet keine Unterstützung für andere Frameworks oder Technologien.

3. Analyse und Konzeption

## 3.1 IST-Analyse

### 3.1.1 Herausforderung und Problemstellung

Im Rahmen der IST-Analyse für Synth UI wurden mehrere Herausforderungen und Problemstellungen identifiziert, die die Entwicklung und Implementierung der Anwendung beeinflussen:

* **Komplexität der UI-Entwicklung:** Die manuelle Erstellung von Benutzeroberflächen in React kann zeitaufwendig und fehleranfällig sein, insbesondere bei komplexen Anforderungen und sich häufig ändernden Designs.
* **Variabilität der Nutzerbedürfnisse:** Die Anforderungen der Nutzer an UI-Komponenten sind oft divers und variieren je nach Projekt, was die Notwendigkeit einer flexiblen und anpassbaren Lösung unterstreicht.
* **Mangel an Benutzerfreundlichkeit:** Viele bestehende Tools zur UI-Entwicklung sind nicht intuitiv und erfordern umfangreiche technische Kenntnisse, was die Nutzung durch nicht-technische Nutzer einschränkt.
* **Integration in bestehende Workflows:** Die Herausforderung, Synth UI nahtlos in bestehende Entwicklungsprozesse und Tools zu integrieren, um die Produktivität der Teams zu erhöhen.
* **Einschränkungen bei der Wiederverwendbarkeit:** In der aktuellen Praxis werden UI-Komponenten oft nicht effektiv wiederverwendet, was zu redundanter Arbeit und inkonsistenten Designs führt.

Diese Herausforderungen bilden die Grundlage für die Entwicklung von Synth UI und leiten die Lösungsansätze, die in der weiteren Analyse und Konzeption behandelt werden.

### 3.1.2 Stakeholder-Analyse

Die Stakeholder-Analyse von Synth UI hat zum Ziel, die relevanten Interessengruppen zu identifizieren und zu klassifizieren, die Einfluss auf das Projekt ausüben oder von dessen Ergebnissen betroffen sind.

* **Entwicklerteams:**
  + *Interessen:* Effizienzsteigerung bei der Erstellung von UI-Komponenten, Zugriff auf anpassbare und wiederverwendbare Komponenten.
  + *Erwartungen:* Intuitive Nutzung von Synth UI, die Integration in bestehende Workflows und Tools.
* **Projektmanager:**
  + *Interessen:* Einhaltung von Projektfristen, Kostenkontrolle und Risikomanagement.
  + *Erwartungen:* Eine Lösung, die die Produktivität erhöht und den Ressourcenaufwand für die UI-Entwicklung minimiert.
* **Endanwender:**
  + *Interessen:* Benutzerfreundliche und ansprechende Benutzeroberflächen, die ihre Bedürfnisse erfüllen.
  + *Erwartungen:* Eine verbesserte Nutzererfahrung durch konsistente und leicht zugängliche UI-Elemente.
* **Management und Entscheidungsträger:**
  + *Interessen:* Strategische Ausrichtung und Rentabilität des Projekts.
  + *Erwartungen:* Nachweisbare Effizienzgewinne und Wettbewerbsvorteile durch die Einführung von Synth UI.
* **Technische Support-Teams:**
  + *Interessen:* Einfache Wartung und Unterstützung der Anwendung.
  + *Erwartungen:* Klar dokumentierte Schnittstellen und ein robustes Fehlermanagement.

### 3.1.3 Nutzerprofil und Personas

Im Rahmen der technischen Dokumentation von Synth UI werden die Nutzerprofile und Personas definiert, um die Zielgruppen und deren spezifische Anforderungen zu erfassen:

**Nutzerprofile:**

1. **Frontend-Entwickler:**
   * *Alter:* 25–40 Jahre
   * *Erfahrung:* 3–10 Jahre in der Webentwicklung, mit spezifischem Fokus auf React.
   * *Bedürfnisse:* Effiziente Erstellung und Anpassung von UI-Komponenten, Wiederverwendbarkeit bestehender Elemente.
   * *Ziele:* Zeitersparnis bei der Entwicklung und Förderung konsistenter Designs.
2. **UI/UX-Designer:**
   * *Alter:* 25–35 Jahre
   * *Erfahrung:* 2–8 Jahre in der Gestaltung von Benutzeroberflächen.
   * *Bedürfnisse:* Zugriff auf anpassbare Komponenten, die den Designrichtlinien entsprechen.
   * *Ziele:* Schnelle Umsetzung von Designideen und -Prototypen.
3. **Projektmanager:**
   * *Alter:* 30–50 Jahre
   * *Erfahrung:* 5–15 Jahre in der Projektleitung, häufig im IT- oder Softwareentwicklungsbereich.
   * *Bedürfnisse:* Transparente Kommunikation über den Fortschritt der Entwicklung, Einhaltung von Zeit- und Budgetvorgaben.
   * *Ziele:* Effiziente Ressourcenallokation und termingerechte Projektabwicklung.

**Personas:**

1. **Max, der Frontend-Entwickler:**
   * *Hintergrund:* Max ist ein leidenschaftlicher Entwickler, der gerne mit den neuesten Technologien arbeitet. Er hat viel Erfahrung mit React und möchte seine Effizienz steigern.
   * *Motivation:* Max sucht nach einem Tool, das ihm hilft, die Entwicklungszeit zu verkürzen und gleichzeitig die Qualität seiner Arbeit zu verbessern.
2. **Sara, die UI/UX-Designerin:**
   * *Hintergrund:* Sara hat einen Hintergrund in Grafikdesign und spezialisiert sich auf die Gestaltung von benutzerfreundlichen Oberflächen. Sie benötigt eine schnelle Möglichkeit, ihre Entwürfe in funktionierende Komponenten zu verwandeln.
   * *Motivation:* Sara möchte sicherstellen, dass ihre Designs korrekt umgesetzt werden, ohne viel Zeit mit der Codierung verbringen zu müssen.
3. **Tom, der Projektmanager:**
   * *Hintergrund:* Tom hat mehrere Projekte im Bereich Softwareentwicklung geleitet. Er ist daran interessiert, wie Synth UI die Zusammenarbeit zwischen Entwicklern und Designern verbessern kann.
   * *Motivation:* Tom sucht nach Wegen, die Effizienz und Produktivität seines Teams zu steigern, während er gleichzeitig den Überblick über den Projektfortschritt behält.

### 3.1.4 Lösungsansatz PoC

Der Lösungsansatz für Synth UI basiert auf einem **Proof of Concept** (PoC), der darauf abzielt, die Machbarkeit und den Nutzen der Anwendung zu demonstrieren. Folgende Schlüsselkomponenten und Strategien wurden im Rahmen des PoC verfolgt:

* **Integration von LLMs:** Der PoC nutzt verschiedene Large Language Models (LLMs), um die automatische Generierung von React-Komponenten aus natürlichen Spracheingaben zu ermöglichen. Dies zeigt, wie effektiv die Anwendung komplexe UI-Anforderungen umsetzen kann.
* **Wissensdatenbank:** Eine zentrale Wissensdatenbank, bestehend aus fünf UI-Bibliotheken mit über 100 Komponenten, wird implementiert. Diese Bibliothek dient als Basis für die Generierung von Komponenten und stellt sicher, dass gängige Designrichtlinien eingehalten werden.
* **Benutzeroberfläche:** Eine intuitive Benutzeroberfläche wird entwickelt, um die Interaktion mit der Anwendung zu erleichtern. Der Fokus liegt auf einer einfachen und klar strukturierten Eingabemaske, die es den Nutzern ermöglicht, Anforderungen schnell und unkompliziert zu formulieren.
* **Prototyping:** Im Rahmen des PoC werden Prototypen erstellt, um die generierten Komponenten in realen Anwendungsszenarien zu testen. Dies ermöglicht eine frühe Validierung der Funktionalität und Benutzererfahrung.
* **[ausstehend] Feedback-Mechanismus:** Ein System zur Erfassung von Nutzerfeedback wird eingerichtet, um kontinuierliche Verbesserungen basierend auf den Erfahrungen der Benutzer zu ermöglichen. Dies ist entscheidend, um die Benutzerfreundlichkeit und Funktionalität weiter zu optimieren.
* **Technische Validierung:** Der PoC beinhaltet eine technische Validierung der erstellten Komponenten hinsichtlich ihrer Leistungsfähigkeit, Stabilität und Anpassungsfähigkeit an unterschiedliche Anforderungen.

Durch diesen Lösungsansatz soll der PoC die Grundlage für eine mögliche weitere Entwicklung von Synth UI schaffen und zeigen, wie die Anwendung den Entwicklungsprozess für React-Komponenten optimieren kann.

## 3.2 SOLL-Prozess der KI-Unterstützung

### 3.2.1 Datenansicht

Synth UI bezieht sich auf die strukturierte Darstellung und den Zugriff auf Informationen, die für die Generierung und Verwaltung von React-Komponenten notwendig sind. Die wesentlichen Aspekte der Datenansicht umfassen:

* **Datenstruktur:**
  + Die Wissensdatenbank umfasst eine Sammlung von über 100 UI-Komponenten aus fünf verschiedenen UI-Bibliotheken. Jede Komponente wird durch Attribute wie Name, Typ, Eigenschaften, Styling-Optionen und Beispiele beschrieben. Diese Struktur ermöglicht eine einfache und effiziente Abfrage durch das KI-Modell.
* **Zugriffsmechanismen:**
  + Die Anwendung stellt AI-Agents zur Verfügung, die eine flexible Abfrage der Wissensdatenbank ermöglichen. Nutzer können gezielt nach bestimmten Komponenten suchen oder Anforderungen formulieren, die das LLM interpretieren kann, um geeignete Lösungen vorzuschlagen.
* **Datenquellen:**
  + Die Daten für die Komponenten stammen aus externen UI-Bibliotheken, die kontinuierlich aktualisiert und gepflegt werden. Dies gewährleistet, dass die Anwendung verifizierte und relevante Informationen bereitstellt. Notiz: Der aktuelle Stand ist jeweils nur eine Momentaufnahme und wird nicht synchronisiert mit Änderungen in der Bibliothek.
* **[ausstehend] Versionierung:**
  + Die Datenansicht umfasst ein Versionskontrollsystem, das es ermöglicht, Änderungen an den Komponenten nachzuvollziehen. Dies ist besonders wichtig für die Wartung und Weiterentwicklung der Anwendung, da frühere Versionen jederzeit zugänglich sind.

### 3.2.2 Nutzersicht

Der Fokus von Synth UI liegt darauf, den Nutzern eine intuitive und effiziente Möglichkeit zur Erstellung von React-Komponenten durch natürliche Sprache zu bieten. Die wichtigsten Aspekte der Nutzersicht umfassen:

* **Benutzeroberfläche:**
  + Die Anwendung bietet eine klare und benutzerfreundliche Oberfläche, die es Nutzern ermöglicht, Anforderungen schnell und unkompliziert einzugeben. Die Eingabemaske ist so gestaltet, dass sie alle relevanten Informationen erfasst, ohne die Nutzer mit überflüssigen Optionen zu überladen.
* **Eingabemethoden:**
  + Nutzer können ihre Anforderungen in natürlicher Sprache formulieren, was die Einstiegshürde senkt und eine breitere Nutzerbasis anspricht. Die KI analysiert die Eingaben und schlägt passende Lösungen vor.
* **Dokumentation und Hilfestellungen:**
  + Die Anwendung bietet Zugriff auf umfassende Dokumentationen und Tutorials, um Nutzern zu helfen, das volle Potenzial von Synth UI auszuschöpfen. Dies schließt Beispielanwendungen und häufig gestellte Fragen (FAQ) ein.
* **Integration und Export:**
  + Die entwickelten Komponenten können nahtlos in bestehende Entwicklungsumgebungen exportiert werden. Dies erleichtert die Nutzung der generierten Komponenten in laufenden Projekten und spart Zeit im Entwicklungsprozess.

Die Nutzersicht von Synth UI ist darauf ausgelegt, eine positive Benutzererfahrung zu gewährleisten, indem sie sowohl Effizienz als auch Benutzerfreundlichkeit fördert. Durch natürliche Spracheingaben wird die Erstellung von React-Komponenten erheblich vereinfacht.

### 3.2.3 Systemarchitektur

Die Systemarchitektur von Synth UI ist so gestaltet, dass die gesamte KI-Logik auf dem Server ausgeführt wird. Dies gewährleistet eine zentrale Steuerung und effiziente Verarbeitung von Benutzeranfragen. Die Architektur besteht aus mehreren Schlüsselelementen:

* **Zentraler AI Core:**
  + Der AI Core bildet das Herzstück der Anwendung und steuert alle KI-gestützten Funktionen. Er ist verantwortlich für die Verarbeitung der natürlichen Spracheingaben der Nutzer und die Koordination der verschiedenen Agenten im Workflow. Der AI Core ermöglicht eine konsistente und optimierte Generierung von React-Komponenten.
* **Agents:**
  + Mehrere spezialisierte Agenten übernehmen unterschiedliche Rollen im Workflow. Jeder Agent ist darauf ausgelegt, spezifische Aufgaben zu erfüllen, wie beispielsweise die Analyse von Nutzeranfragen, die Generierung von Code oder die Verwaltung von Feedback. Diese Aufteilung ermöglicht eine effektive Nutzung der Ressourcen und eine schnellere Verarbeitung von Benutzeranfragen.
* **Client-Server-Architektur:**
  + Die Anwendung folgt einem Client-Server-Modell, bei dem der Client die Benutzeroberfläche bereitstellt und der Server die gesamte KI-Logik und Datenverarbeitung übernimmt. Diese Trennung ermöglicht eine verbesserte Wartbarkeit und Skalierbarkeit.
* **Sicherheitsmechanismen:**
  + Die Architektur umfasst Sicherheitsmaßnahmen wie Authentifizierung und Autorisierung, um den Zugriff auf sensible Daten zu steuern. Verschlüsselung und Sicherheitsprotokolle schützen die Datenübertragung zwischen Client und Server.

Diese Systemarchitektur ermöglicht Synth UI, eine leistungsstarke und benutzerfreundliche Lösung zur Erstellung von React-Komponenten zu implementieren, indem die KI-Logik zentralisiert und durch spezialisierte Agenten unterstützt wird.

## 3.3 Technische Verifikation des KI-Einsatzes

### 3.3.1 Datengrundlage und -qualität

Synth UI setzt auf der Beschaffung, Verwaltung und Sicherstellung der Datenqualität, um optimale Ergebnisse zu erzielen. Die wichtigsten Aspekte sind:

* **Wissensdatenbank:**
  + Die Wissensdatenbank umfasst über 100 UI-Komponenten, die aus fünf verschiedenen UI-Bibliotheken stammen. Jede Komponente ist umfassend dokumentiert und enthält relevante Attribute wie Name, Typ, Eigenschaften und Styling-Optionen. Diese strukturierte Datenbasis ermöglicht eine präzise Analyse durch den AI Core.
* **Zusätzliche Datensätze:**
  + Für das Modell, das für Synth UI entwickelt wird (derzeit noch in der Fertigstellung), wurden zwei Datensätze erstellt, die auf zehn Community-Repositories im Bereich Next.js/React basieren. Diese Datensätze tragen zur Verbesserung der Sprachverarbeitung und der Generierung von React-Komponenten bei.

### 3.3.2 Einsatzbereich und Limitationen

Synth UI bietet vielfältige Einsatzmöglichkeiten, bringt jedoch auch einige Limitationen mit sich, die berücksichtigt werden sollten. Die wesentlichen Punkte sind:

* **Einsatzbereich:**
  + **Generierung, Prototyping und Anpassung von React-Komponenten:** Nutzer können durch natürliche Spracheingaben benutzerdefinierte React-Komponenten erstellen, was den Entwicklungsprozess beschleunigt.
* **Limitationen:**
  + **Eingeschränkte Kontextverarbeitung und Abhängigkeit von Datenqualität:** Die Qualität der generierten Komponenten hängt von der Klarheit der Eingaben ab. Komplexe Anforderungen können zu suboptimalen Ergebnissen führen.

## 3.4 Erfolgskriterien und KPIs

### 3.4.1 Quantitative Merkmale

Synth UI versucht durch numerische Indikatoren, die zur Analyse der Anwendung und ihrer Leistung verwendet werden können, Erfolg zu messen. Die wichtigsten quantitativen Merkmale im Zusammenhang mit Erfolgskriterien und KPIs sind:

* **Nutzeranzahl:**
  + **Aktive Nutzer:** Die Anzahl der wöchentlich oder monatlich aktiven Nutzer, die die Anwendung regelmäßig verwenden.
  + **Neuregistrierungen:** Die Anzahl neuer Nutzer, die sich in einem bestimmten Zeitraum registrieren.
* **Generierungszeiten:**
  + **Durchschnittliche Generierungszeit:** Die durchschnittliche Zeit (in Sekunden oder Minuten), die benötigt wird, um eine React-Komponente zu generieren.
  + **90. Perzentil der Generierungszeit:** Die Zeit, die 90 % der Generierungsanfragen nicht überschreitet, um herauszufinden, wie viele Nutzer schnellere Reaktionszeiten erwarten können.
* **Fehlerrate:**
  + **Anteil der fehlerhaften Komponenten:** Der Prozentsatz der generierten Komponenten, die nicht den Anforderungen entsprechen oder Fehler enthalten. Eine Benchmark könnte unter 5 % liegen.
  + **Anzahl der Anpassungen:** Die durchschnittliche Anzahl der Anpassungen, die pro generierter Komponente erforderlich sind, um den Anforderungen der Nutzer gerecht zu werden.

Diese Merkmale liefern objektive Daten, die zur Überwachung der Leistung von Synth UI und zur Identifizierung von Verbesserungsmöglichkeiten genutzt werden können.

### 3.4.2 Qualitative Merkmale

Qualitative Merkmale spielen eine zentrale Rolle bei der Bewertung der Benutzererfahrung und der Effektivität von Synth UI. Die wichtigsten Feststellungsmerkmale sind:

* **Nutzererfahrung (UX):**
  + Eine umfassende Bewertung der Benutzeroberfläche hinsichtlich ihrer Intuitivität und Benutzerfreundlichkeit, sowie der Gesamtästhetik und des Designs.
* **Qualität der generierten Komponenten:**
  + Subjektive Bewertungen der Funktionalität und Ästhetik der generierten Komponenten.

Diese qualitativen Merkmale liefern essentielle Informationen zur kontinuierlichen Verbesserung von Synth UI und helfen die Anwendung an die Bedürfnisse der Nutzer anzupassen.

### 3.4.3 Testszenarien und Benchmarks

Testszenarien und Benchmarks sind wichtig für die Evaluierung der Leistungsfähigkeit und UX von Synth UI. Die wesentlichen Aspekte umfassen:

* **Testszenarien:**
  + Tests zur Fehlerbehandlung, um zu evaluieren, wie die Anwendung mit ungültigen Eingaben umgeht und welche Mechanismen zur Problemlösung bereitgestellt werden.
* **Benchmarks:**
  + Leistungsbenchmarks, die die Generierungszeiten und Ressourcennutzung von Synth UI mit ähnlichen Anwendungen oder Vorgängerversionen vergleichen, um die Effizienz zu maximieren.
  + Integrationstests, die die Erfolgsquote der generierten Komponenten in unterschiedlichen Entwicklungsumgebungen und Frameworks messen.

Diese Testszenarien und Benchmarks können für die kontinuierliche Verbesserung von Synth UI hilfreich sein und zeigen, ob die Anwendung die Erwartungen der Nutzer erfüllt und dabei hohe Qualitätsstandards einhält.

4. Technische Spezifikation

## 4.1 Softwarearchitektur & Datenmodell

Die Softwarearchitektur von Synth UI folgt einem modularen Ansatz und besteht aus einem zentralen AI Core, der die KI-Logik steuert, und mehreren spezialisierten Agents, die spezifische Aufgaben im Workflow übernehmen. Das Frontend bietet eine intuitive Benutzeroberfläche, die über RESTful APIs (NextJS Server Actions) mit dem Backend kommuniziert.

Das Datenmodell von Synth UI nutzt einen Redis Key-Value Storage, der ausschließlich zur Speicherung von Chats dient. Alle anderen Daten, wie Informationen zu den verfügbaren UI-Komponenten, Benutzerprofile und Projektinformationen, sind ausgelagert und werden über externe Systeme verwaltet. Dieser Ansatz ermöglicht eine schnelle und effiziente Handhabung von Chat-Daten, während die restlichen Informationen flexibel und skalierbar in separaten Datenquellen gespeichert werden.

5. Implementierung

## 5.1 Code Standards

Die wichtigsten Richtlinien für Code Qualität für Synth UI umfassen:

* **Linting:** Verwendung von **ESLint** zur Durchsetzung eines konsistenten JavaScript-Stils, einschließlich der Benennungskonventionen für Variablen und Funktionen sowie der Formatierung von Code (z. B. Einrückungen und Leerzeilen).
* **Dokumentation:** Jeder Codeabschnitt (zumindest für Utility Funktionen) muss ausreichend dokumentiert werden, einschließlich Funktionen, Klassen und Module. **JSDoc** wird verwendet, um die Dokumentation im Code zu erstellen und zu verwalten.
* **Versionierung:** Alle Änderungen am Code sind im [Git-Repository für Synth UI](https://github.com/Julian-AT/synth-ui), mit klaren Commit-Nachrichten, die den Zweck der Änderungen beschreiben und Einhaltung von commit-lint, dokumentiert.

## 5.2 CI/CD & DevOps

Das Synth UI-Projekt nutzt eine Continuous Integration/Continuous Deployment (CI/CD)-Pipeline, die auf Vercel implementiert ist. Bei jedem Commit im Repository wird automatisch ein Production Build erstellt und der Code in die Zielumgebung geladen. Dieser automatisierte Prozess gewährleistet, dass alle Änderungen schnell und effizient bereitgestellt werden, wodurch die Entwicklungszyklen verkürzt und die Qualität des Codes kontinuierlich verbessert werden. Die Integration von Tests in den CI/CD-Prozess stellt sicher, dass neue Änderungen keine bestehenden Funktionen beeinträchtigen und die Anwendung stets stabil bleibt. Außerdem bietet Vercel viele Developer Experience Features wie Instant Rollbacks oder Log Retention