

Blatt 4

Software Engineering 2

Luis Staudt

Aufgabe 1

1. Ziele der Spezifikationen:

- **SyRS (System Requirements Specification):** Identifiziert technische Anforderungen für das System-of-Interest und beschreibt, was das System aus technischer Sicht tun soll. Dient als Brücke zwischen Auftraggeber und technischer Community, indem sie alle Eingaben, Ausgaben und erforderlichen Beziehungen zwischen diesen vollständig beschreibt.
- **SRS (Software Requirements Specification):** Spezifiziert detailliert die Anforderungen für ein bestimmtes Softwareprodukt oder Programm in einer spezifischen Umgebung. Definiert alle erforderlichen Fähigkeiten, dokumentiert Bedingungen und Einschränkungen unter denen die Software arbeiten muss.

2. Notwendigkeit:

- **SyRS:** Wird benötigt, wenn ein System entwickelt wird, das mehrere Komponenten umfasst und eine übergeordnete technische Beschreibung erforderlich ist.
- **SRS:** Wird benötigt, wenn spezifisch ein Softwareprodukt entwickelt wird. Bei größeren Systemen erweitert die SRS die im SyRS definierten softwarebezogenen Anforderungen und stimmt mit diesen überein.

2. Vergleich zwischen SyRS und SRS

Aspekt	System Requirements Specification (SyRS)	Software Requirements Specification (SRS)
Fokus	Gesamtsystem (Hardware, Software, Mensch)	Speziell Softwarekomponenten
Umfang	Systemebene, kann mehrere Komponenten umfassen	Fokussiert auf Softwareprogramm oder -set
Detaillierungsgrad	Übergeordnete Systemanforderungen	Detaillierte Softwareanforderungen
Spezifische Inhalte	Physikalische Eigenschaften, Umgebungsbedingungen, Systemsicherheit	Logische Datenbankanforderungen, Softwareattribute
Gemeinsamkeiten	Beide enthalten: funktionale Anforderungen, Schnittstellen, Leistungsanforderungen, Benutzeranforderungen und Verifizierungsansätze	
Verantwortung	Anforderer und Entwicklerteam	Lieferant, Auftraggeber oder beide

Tabelle 1: Vergleich zwischen SyRS und SRS

Aufgabe 2

Revisionshistorie

Datum	Version	Autor	Beschreibung

Definitionen

Definitionen für Wörter oder Ausdrücke, die eine besondere Bedeutung über normale Wörterbuchdefinitionen hinaus haben.

Begriff	Definition

Akronyme und Abkürzungen

Ausschreiben oder Definition aller im Dokument verwendeten Akronyme und Abkürzungen

Akronym/Abkürzung	Definition

Einleitung

Zweck

Beschreibung des Zwecks der zu spezifizierenden Software. Erläuterung, wofür diese Spezifikation verwendet wird und an wen sie sich richtet.

Umfang

Beschreibung des Umfangs der betreffenden Software. Identifikation der Software-Produkte nach Namen. Erklärung, was die Software-Produkte tun werden. Beschreibung der Anwendung der spezifizierten Software.

Produktperspektive

Definition der Beziehung des Systems zu anderen verwandten Produkten. Wenn das Produkt Teil eines größeren Systems ist, Beschreibung der Beziehung. Identifikation der Schnittstellen zwischen dem Produkt und dem größeren System.

Systemschnittstellen

Auflistung jeder Systemschnittstelle und Identifikation der Funktionalität.

Benutzerschnittstellen

Spezifikation der logischen Merkmale jeder Schnittstelle zwischen Software und Benutzern.

Betriebsabläufe

Spezifikation der normalen und speziellen Operationen, die vom Benutzer benötigt werden.

Produktfunktionen

Zusammenfassung der Hauptfunktionen, die die Software ausführen wird. Organisation der Funktionen, um die Liste für den Auftraggeber verständlich zu machen. Verwendung textueller oder grafischer Methoden, um verschiedene Funktionen und Beziehungen darzustellen.

Benutzermerkmale

Beschreibung der allgemeinen Merkmale der beabsichtigten Benutzer. Bildungsniveau, Erfahrung, technisches Fachwissen, mögliche Behinderungen und andere Faktoren.

Einschränkungen

Allgemeine Beschreibung von Faktoren, die die Optionen des Entwicklers einschränken. Regulatorische Richtlinien, Hardware-Beschränkungen, Schnittstellen zu anderen Anwendungen, Parallelbetriebs-Anforderungen, Sicherheits- und Schutzanforderungen.

Annahmen und Abhängigkeiten

Auflistung von Faktoren, die die in der SRS angegebenen Anforderungen beeinflussen. Annahmen über die Verfügbarkeit von Ressourcen, Komponenten usw. Abhängigkeiten von externen Faktoren.

Spezifische Anforderungen

Externe Schnittstellen

Definition aller Ein- und Ausgaben des Softwaresystems. Name des Elements, Beschreibung des Zwecks, Quelle der Eingabe oder Ziel der Ausgabe, Gültigkeitsbereich, Genauigkeit oder Toleranz, Maßeinheiten, Timing, Datenformate.

Funktionen

Definition der grundlegenden Aktionen der Software. Gültigkeitsprüfungen der Eingaben, genaue Abfolge von Operationen, Reaktionen auf abnormale Situationen, Auswirkung von Parametern, Beziehung von Ausgaben zu Eingaben.

Qualitätsanforderungen

Definition von Gebrauchstauglichkeits- und Qualitätsanforderungen. Messbare Kriterien für Effektivität, Effizienz, Zufriedenheit und Vermeidung von Schäden.

Leistungsanforderungen

Spezifikation sowohl statischer als auch dynamischer numerischer Anforderungen. Statische Anforderungen (z.B. Anzahl von Terminals, Benutzern, zu verarbeitende Daten). Dynamische Anforderungen (z.B. Transaktionen/Aufgaben pro Zeiteinheit).

Design einschränkungen

Spezifikation von Einschränkungen für das Systemdesign, die durch externe Standards auferlegt werden. Einhaltung von Standards, Hardware- und Softwarebeschränkungen.

Standardkonformität

Spezifikation von Anforderungen, die sich aus bestehenden Standards oder Vorschriften ergeben. Berichtsformat, Datenbenennung, Buchführungsverfahren, Prüfpfadverfolgung.

Softwaresystemattribute

Spezifikation erforderlicher Attribute des Softwareprodukts.

Zuverlässigkeit

Spezifikation von Faktoren, die erforderlich sind, um die erforderliche Zuverlässigkeit zu gewährleisten.

Verfügbarkeit

Spezifikation von Faktoren, die erforderlich sind, um ein definiertes Verfügbarkeitsniveau zu garantieren.

Sicherheit

Spezifikation von Anforderungen zum Schutz der Software vor versehentlichem oder böswilligem Zugriff, Nutzung, Modifikation, Zerstörung oder Offenlegung.

Wartbarkeit

Spezifikation von Attributen der Software, die sich auf die Wartungsfreundlichkeit beziehen.

Portabilität

Spezifikation von Attributen, die sich auf die Übertragbarkeit der Software in andere Umgebungen beziehen.

Verifizierung

Bereitstellung der Verifizierungsansätze und -methoden, die zur Qualifizierung der Software geplant sind. Verifizierungsansätze für jede Anforderung, Testmethoden, Validierungskriterien.

Aufgabe 3

Revisionshistorie

Datum	Version	Autor	Beschreibung
18. April 2025	1.0	Luis Staudt	Initiale Version

Definitionen

Begriff	Definition
Bloom'sche Taxonomie	Hierarchisches System zur Klassifizierung von Lernzielen in sechs Ebenen: erinnern, verstehen, anwenden, analysieren, bewerten, erschaffen
Geschlossene Aufgabe	Aufgabentyp mit vorgegebenen Antwortmöglichkeiten
Offene Aufgabe	Aufgabentyp ohne vorgegebene Antwortmöglichkeiten
Modul	Lehrveranstaltungseinheit im akademischen Kontext

Referenzen

1. Fallstudie: Generierung von Klausuren aus bestehender Klausuraufgabensammlung

Akronyme und Abkürzungen

Akronym/Abkürzung	Definition
SRS	Software Requirements Specification
DB	Datenbank

Einleitung

Zweck

Diese Softwareanforderungsspezifikation (SRS) beschreibt die funktionalen und nicht-funktionalen Anforderungen an den Klausurgenerator, der zur Erstellung und Verwaltung von Klausuraufgaben und zur Zusammenstellung von Klausuren dient. Dieses Dokument richtet sich an das Entwicklungsteam sowie an die zukünftigen Nutzer des Systems.

Umfang

Der Klausurgenerator soll Dozenten ermöglichen, Klausuraufgaben zu erstellen, zu verwalten und zu kategorisieren, sowie aus diesen Aufgaben Klausuren zusammenzustellen. Die Software soll folgende Hauptfunktionen bieten:

- Eingabe und Speicherung von Klausuraufgaben mit definierten Attributen in einer Datenbank
- Speicherung von Musterlösungen oder richtigen Lösungen
- Zuordnung von Aufgaben zu Lehrveranstaltungen/Modulen
- Zusammenstellung von Klausuren nach definierten Kriterien
- Druck von einzelnen Aufgaben und vollständigen Klausuren

Produktperspektive

Der Klausurgenerator ist ein eigenständiges System, das von Dozenten zur Klausurerstellung und -verwaltung genutzt wird. Es ist nicht in bestehende Systeme integriert.

Systemschnittstellen

Der Klausurgenerator benötigt keine externen Systemschnittstellen in der ersten Version.

Benutzerschnittstellen

Die Benutzerschnittstelle soll intuitiv und benutzerfreundlich sein und folgende Bereiche umfassen:

- Eingabeformulare für Klausuraufgaben mit allen Attributen
- Verwaltungsbereich für bestehende Aufgaben
- Filterbereich zur Auswahl von Aufgaben nach verschiedenen Kriterien
- Klausurzusammenstellungsbereich
- Druck- und Exportfunktionen

Produktfunktionen

Die Hauptfunktionen des Klausurgenerators sind:

- Verwaltung von Klausuraufgaben (Erstellen, Bearbeiten, Löschen)
- Kategorisierung von Aufgaben nach Bloom'scher Taxonomie, Format und Typ

- Zusammenstellung von Klausuren aus vorhandenen Aufgaben
- Druck von Aufgaben und Klausuren

Benutzermerkmale

Die primären Nutzer sind Dozenten an Bildungseinrichtungen, die regelmäßig Klausuren erstellen müssen. Es wird erwartet, dass die Nutzer grundlegende Computerkenntnisse besitzen und mit pädagogischen Konzepten wie der Bloom'schen Taxonomie vertraut sind.

Einschränkungen

- Die Software soll auf gängigen Betriebssystemen lauffähig sein
- Die Benutzerschnittstelle soll in deutscher Sprache sein

Annahmen und Abhängigkeiten

- Es wird angenommen, dass die Nutzer über ein geeignetes Computersystem verfügen
- Es wird angenommen, dass die Nutzer Grundkenntnisse über Klausurerstellung haben

Spezifische Anforderungen

Externe Schnittstellen

Benutzerschnittstellen

- Die Benutzerschnittstelle soll übersichtlich und intuitiv bedienbar sein
- Formulare zur Eingabe von Aufgaben sollen alle erforderlichen Felder enthalten
- Die Zusammenstellung von Klausuren soll durch einen Assistenten unterstützt werden

Funktionen

Aufgabenverwaltung

- Die Software soll das Erstellen, Bearbeiten und Löschen von Klausuraufgaben ermöglichen
- Jede Aufgabe soll folgende Attribute haben: Name, Aufgabentext, Antwortmöglichkeiten (bei geschlossenen Aufgaben), geschätzte Zeit, Modulzugehörigkeit
- Jede Aufgabe soll nach folgenden Kriterien kategorisiert werden:
 - Bloom'sche Taxonomie (Level 1-6)

- Aufgabenformat (offen/geschlossen)
- Typ der geschlossenen Aufgabe
- Zu jeder Aufgabe soll eine Musterlösung oder richtige Lösung gespeichert werden

Klausurzusammenstellung

- Die Software soll die Zusammenstellung von Klausuren aus vorhandenen Aufgaben ermöglichen
- Aufgaben sollen nach verschiedenen Kriterien filterbar sein
- Die Zusammenstellung soll gespeichert werden können

Druck und Export

- Die Software soll den Druck einzelner Aufgaben ermöglichen
- Die Software soll den Druck vollständiger Klausuren ermöglichen

Leistungsanforderungen

- Die Software soll auch bei einer großen Anzahl von Aufgaben performant bleiben
- Die Reaktionszeit bei Filteroperationen soll angemessen sein

0.0.1 *Softwaresystemattribute

Zuverlässigkeit

Die Software soll stabil laufen und keine Datenverluste verursachen.

Verfügbarkeit

Die Software soll als lokale Anwendung jederzeit verfügbar sein.

Sicherheit

Der Zugriff auf die Software soll durch ein Anmeldesystem geschützt sein.

Wartbarkeit

Die Software soll über Import- und Exportfunktionen für die Datensicherung verfügen.

Verifizierung

Für jede Anforderung sollen Testfälle definiert werden, um die korrekte Implementierung zu überprüfen.

Anhänge

Konzeptionelle Modellierung

User Stories mit Akzeptanzkriterien

*User Story 1 **Als** Dozent **möchte ich** Aufgaben mit allen notwendigen Attributen und Kategorisierungen in der Datenbank speichern können, **damit** ich eine umfangreiche Sammlung an Klausuraufgaben erstellen kann.

Akzeptanzkriterien:

- Alle definierten Attribute können eingegeben werden
- Bloom'sche Taxonomie Level kann ausgewählt werden
- Aufgabenformat kann festgelegt werden
- Bei geschlossenen Aufgaben kann der Typ spezifiziert werden
- Die Lösung oder Musterlösung kann gespeichert werden

*User Story 2 **Als** Dozent **möchte ich** Klausuren aus vorhandenen Aufgaben nach definierten Kriterien zusammenstellen können, **damit** ich effizient Prüfungen erstellen kann.

Akzeptanzkriterien:

- Aufgaben können nach Modul gefiltert werden
- Aufgaben können nach Bloom'scher Taxonomie gefiltert werden
- Aufgaben können nach Format und Typ gefiltert werden
- Die Gesamtzeit der Klausur wird angezeigt
- Die ausgewählten Aufgaben können als Klausur gespeichert werden

Anwendungsfälle

*Anwendungsfall 1: Aufgabe erstellen und speichern

*Anwendungsfall 2: Klausur zusammenstellen

 ||X|

Name: Aufgabe erstellen und speichern

Akteur: Dozent

Beschreibung: Der Dozent erstellt eine neue Klausuraufgabe und speichert sie mit allen erforderlichen Attributen und Kategorisierungen in der Datenbank.

Vorbedingung: Der Dozent ist im System angemeldet.

Standardablauf:

1. Der Dozent wählt die Option "Neue Aufgabe erstellen".
2. Das System zeigt ein Formular zur Eingabe aller Attribute.
3. Der Dozent gibt den Namen, Aufgabentext und ggf. Antwortmöglichkeiten ein.
4. Der Dozent wählt die Zugehörigkeit zu einem Modul.
5. Der Dozent schätzt die Bearbeitungszeit.
6. Der Dozent ordnet die Aufgabe einem kognitiven Level der Bloom'schen Taxonomie zu.
7. Der Dozent legt das Aufgabenformat fest (offen/geschlossen).
8. Bei geschlossenen Aufgaben wählt der Dozent den Typ.
9. Der Dozent gibt die Lösung oder Musterlösung ein.
10. Der Dozent speichert die Aufgabe.
11. Das System bestätigt die erfolgreiche Speicherung.

Alternative Abläufe: Wenn der Dozent nicht alle erforderlichen Felder ausfüllt, zeigt das System eine Fehlermeldung.

Nachbedingung: Die neue Aufgabe ist in der Datenbank gespeichert.

Architekturdesign auf konzeptioneller Ebene

▪ Präsentationsschicht:

- Benutzeroberfläche für Dozenten
- Formulare zur Eingabe von Aufgaben
- Übersichten und Filtermöglichkeiten
- Druckvorschau

▪ Anwendungsschicht:

- Aufgabenverwaltung
- Klausurzusammenstellung
- Filterlogik
- Druckaufbereitung

▪ Datenhaltungsschicht:

- Datenbank für Aufgaben und deren Attribute
- Datenbank für erstellte Klausuren

||X|**Name:** Klausur zusammenstellen

Akteur: Dozent

Beschreibung: Der Dozent stellt aus vorhandenen Aufgaben eine Klausur zusammen.

Vorbedingung: Der Dozent ist im System angemeldet und es sind Aufgaben in der Datenbank vorhanden.

Standardablauf:

1. Der Dozent wählt die Option "Klausur erstellen".
2. Das System zeigt die Filtermöglichkeiten für Aufgaben.
3. Der Dozent wählt das gewünschte Modul.
4. Der Dozent filtert nach weiteren Kriterien (Bloom'sche Taxonomie, Format, Typ).
5. Das System zeigt die passenden Aufgaben.
6. Der Dozent wählt Aufgaben aus und fügt sie zur Klausur hinzu.
7. Das System berechnet die Gesamtzeit der Klausur.
8. Der Dozent speichert die zusammengestellte Klausur.
9. Der Dozent kann die Klausur ausdrucken oder als Datei exportieren.

Alternative Abläufe: Wenn keine passenden Aufgaben vorhanden sind, erhält der Dozent eine entsprechende Meldung.

Nachbedingung: Die Klausur ist zusammengestellt und kann verwendet werden.

– Import/Export-Funktionalität