Julius Beier A20/MaL

Anforderungsprofil - SchoolNotes

1. Einleitung:

SchoolNotes wird im Rahmen eines Schulprojektes im Fach Informatik Systeme entwickelt und soll bis zum 13.03.2023 fertig gestellt werden. Es wird entwickelt um den Schulalltag, für Schüler, die abseits von Word und Co, Digital arbeiten, zu erleichtern und besser mit ihren Mitschriften lernen zu können.

2. Lastenheft Teil:

2.1 Ist-Situation:

Zurzeit schreibe ich meine Mitschriften in einem Simplen Texteditor im Markdown Format, dabei mangelt es jedoch an einigen nützlichen Funktionen.

2.2 Soll-Situation:

SchoolNotes soll ein Programm werden womit Schüler ihre Mitschriften Betriebssystem unabhängig niederschreiben können. Diesen Mitschriften sollte man „Tags“ hinzufügen können, um diese Kategorisieren zu können. Ebenfalls sollte man „Tests“ erstellen können und diesen Lernstoff(Notizen) und/oder Fragen hinzufügen können. Dabei sollten Tests und Fragen auch mit „Tags“ verknüpfbar sein.

Notizen sollten in Dateien gespeichert werden und mithilfe einer Datenbank, welche ohne Server fungiert, verknüpft werden. Die restlichen Daten sollen ebenfalls in der Datenbank gespeichert werden.

Das Design der Benutzeroberfläche sollte simpel gehalten werden um mögliche Ablenkungen während des Unterrichts zu vermeiden.

2.3 Systemanforderungen:

Wie bereits genannt sollte das Programm auf jedem Betriebssystem lauffähig sein und so wenig Ressourcen wie möglich beanspruchen.

2.4 Qualitätsanforderungen (Funktional):

Man sollte Notizen, Tests und Aufgaben so leicht wie möglich hinzufügen können, mithilfe jeweiliger Buttons direkt auf der Startseite. Ebenfalls sollte man diesen Tags und Anmerkungen hinzufügen können. Dabei sollten alle drei ihre eigenen separaten Tags besitzen. Optional sollte man seine Mitschriften exportieren und importieren können.

2.5 Qualitätsanforderungen (Nicht Funktional):

Das Programm sollte so Modular wie möglich gestaltet werden, um spätere Erweiterungen zu erleichtern.

2.6 Abnahmekriterien:

Das Programm sollte vollständig Lauffähig und Fehlerfrei sein. Ebenfalls sollte es mit einem bereits angefertigtem Test Datenbestand versehen sein.

3. Pflichtenheft Teil:

3.1 Software:

Für die Betriebssystem Unabhängigkeit wird Object Pascal mit der IDE Lazarus genutzt. Als Datenbanksystem wird SQLite benutzt, um die Server Unabhängigkeit zu gewährleisten.

Wie Funktionen umgesetzt werden (Notiz => in Datei mit DB verweis, ...)

3.2 Zeitlicher Aspekt:

Vorgangs Nr	Vorgänger	Dauer	Vorgang
1	-	66	Dokumentation anfertigen
2	-	2	Datenbank planen
3	2	7	Benutzeroberfläche planen
4	3	1	Codestandards festlegen
5	4	12	Code Planung (Struktogramme)
6	4,5	10	Benutzeroberfläche implementieren
7	6	20	Funktionalität implementieren
8	7	2	BPMN erstellen
9	8	2	Codestandards überprüfen
10	9	6	Testbetrieb und Bugfixing
11	10	4	Optimierungen finden und implementieren
12	11	6	Datenaufnahme
13	12	1	Abgabe

Vorgang 1: Dokumentation anfertigen

In diesem Vorgang wird die Entwicklung des Projektes dokumentiert und bestimmte Sachen genauer erklärt.

Vorgang 2: Datenbank planen

In diesem Vorgang wird die Datenbank Struktur festgelegt und die einzelnen Beziehungen dargestellt.

Vorgang 3: Benutzeroberfläche Planen

In diesem Vorgang wird die Grafische Oberfläche geplant, um ein möglichst Ablenkungsfreieis Arbeiten zu gewährleisten. Ebenfalls wird die Implementierung dadurch erleichtert.

Vorgang 4: Codestandards festlegen

In diesem Vorgang wird das Aussehen, die Namensgebung und andere Formatierungseinstellungen des Codes festgelegt.

Vorgang 5: Code Planung

In diesem Vorgang wird das Projekt in der Implementierung vorbereitet, um Komplikationen bei der wirklichen Implementierung so weit wie möglich zu umgehen.

Geplant wird mithilfe von Struktogrammen.

Vorgang 6: Benutzeroberfläche implementieren

In diesem Vorgang wird die Benutzeroberfläche in Lazarus implementiert.

Vorgang 7: Funktionalität implementieren

In diesem Vorgang wird die Software mit all ihrer Funktionalität implementiert. Dieser Vorgang bringt dann die erste Version des Software Produkt hervor.

Vorgang 8: BPMN erstellen

In diesem Vorgang wird das benutzen des Programms klar Definiert mithilfe eines Business-Process-Model-Notation Diagramms.

Vorgang 9: Codestandards überprüfen

In diesem Vorgang wird überprüft ob alle Codestandards eingehalten wurden und ob die Code Dokumentation vollständig ist.

Vorgang 10: Testbetrieb und Bugfixing

In diesem Vorgang wird das Programm im Testbetrieb auf Fehler geprüft, welche auch direkt behoben werden.

Vorgang 11: Optimierungen finden und implementieren

In diesem Vorgang wird nach Optimierungsmöglichkeiten im Code gesucht, um die Anforderung der Ressourcenschonung gerecht zu werden.

Vorgang 12: Datenaufnahme

In diesem Vorgang wird der Datenbestand gefüllt, um den Abnahmekriterien gerecht zu werden.

Vorgang 13: Abgabe

In diesem Vorgang wird das Fertige Software Produkt mit dem erarbeiteten Datenbestand abgegeben.