**EDV‑Schulen Wiesau**

**Lastenheft**

NotenManager

**Stand 19.06.2018**

Historie der Dokumentversionen

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Autor | Änderungsgrund / Bemerkungen |
| 0.1 | 19.06.2015 | T. Bareuther | Ersterstellung |
| 0.2 | 24.06.2015 | H. Körner / M. Simmerl | Gruppen eingeteilt |
| 0.3 | 06.07.2015 | H. Körner / M. Simmerl | 3.1.3 und 3.1.4 ergänzt |
| 0.4 | 07.07.2015 | H. Körner / M. Simmerl | 3.1.7 präzisiert und Schülernotenblatt (Scan) ergänzt |
| 0.5 | 13.05.2016 | H. Körner / M. Simmerl | Gruppen eingeteilt |
| 0.6 | 03.06.2016 | H. Körner / M. Simmerl | Lastenheft bzgl. Begriffllichkeiten im Glossar angepasst |
| 0.7 | 10.06.2016 | H. Körner / M. Simmerl | 3.1.6 um konkrete Beispiele ergänzt |
| 0.8 | 19.06.2018 | Körner / Bareuther | Umgestaltung für iterativen Ansatz ab SJ 2017/18 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

[Historie der Dokumentversionen 2](#_Toc453319245)

[Inhaltsverzeichnis 3](#_Toc453319246)

[1 Einleitung 4](#_Toc453319247)

[1.1 Allgemeines 4](#_Toc453319248)

[1.1.1 Zweck und Ziel dieses Dokuments 4](#_Toc453319249)

[1.1.2 Projektbezug 4](#_Toc453319250)

[1.1.3 Abkürzungen 4](#_Toc453319251)

[2 Konzept und Rahmenbedingungen 5](#_Toc453319252)

[2.1 Ziele des Projekts 5](#_Toc453319253)

[2.2 Benutzer / Zielgruppe 5](#_Toc453319254)

[2.3 Systemvoraussetzungen 5](#_Toc453319255)

[2.4 Einbindung in das Gesamtsystem 5](#_Toc453319256)

[3 Beschreibung der Anforderungen 6](#_Toc453319257)

[3.1 Funktionale Anforderungen 6](#_Toc453319258)

[3.1.1 Login 6](#_Toc453319259)

[3.1.2 Stammdaten 6](#_Toc453319260)

[3.1.3 Leistungsarten und Leistungsgruppen 6](#_Toc453319261)

[3.1.4 Notenerfassung 6](#_Toc453319262)

[3.1.5 Protokollierung und Änderungshistorie 7](#_Toc453319263)

[3.1.6 Berechnung eines Vorschlags für Zeugnisnoten 7](#_Toc453319264)

[3.1.7 Berechtigungskonzept 7](#_Toc453319265)

[3.1.8 Fehlzeiten [OPTIONAL] 8](#_Toc453319266)

[3.1.9 Zeugnisbemerkungen [OPTIONAL] 8](#_Toc453319267)

[3.1.10 Druckausgaben 8](#_Toc453319268)

[3.1.11 GUI 8](#_Toc453319269)

[3.2 Nicht-funktionale Anforderungen 8](#_Toc453319270)

[3.2.1 Aufbau 8](#_Toc453319271)

[3.2.2 Fehlerkonzept 8](#_Toc453319272)

[3.2.3 Brauchbarkeit / Benutzerfreundlichkeit 8](#_Toc453319273)

[3.2.4 Zuverlässigkeit/Datensicherheit 9](#_Toc453319274)

[3.2.5 Performance 9](#_Toc453319275)

[3.2.6 Wartbarkeit / Erweiterbarkeit 9](#_Toc453319276)

[3.2.7 Dokumentation 9](#_Toc453319277)

[4 Gruppeneinteilung 10](#_Toc453319278)

[5 Risikobetrachtung 10](#_Toc453319279)

[6 Abnahmekriterien / Testkonzept 10](#_Toc453319280)

[7 Offene Punkte 10](#_Toc453319281)

[8 Hilfreiche Tipps und Hinweise 11](#_Toc453319282)

[Anhang A 12](#_Toc453319283)

# Einleitung

## Allgemeines

### Zweck und Ziel dieses Dokuments

Dieses Lastenheft beschreibt die Anforderungen für ein EDV-System zur Notenerfassung und Verwaltung zum Einsatz an den EDV-Schulen in Wiesau.

### Projektbezug

Durchzuführen ist das Projekt im Rahmen der Ausbildung der Klassen BFI11a bzw. BFI11b im Schuljahr 2017/2018 an den EDV-Schulen Wiesau. Die Projektdokumentation ist bis spätestens 28.09.2018 von jedem Projektteilnehmer individuell zu erstellen.

### Abkürzungen

|  |  |  |
| --- | --- | --- |
| **Begriff** |  | **Erklärung** |
| DB  Digikabu |  | Datenbank  Digitales Klassentagebuch – SW am staatl. beruflichen Schulzentrum Wiesau |
| PDF |  | Portable Document Format |
| GUI |  | Grafisches Benutzer Interface |
| JAVA |  | plattformunabhängige Programmiersprache |
| ODBC |  | Open Database Connectivity (standardisierte Datenbankschnittstelle) |
| Thread  Swing |  | Autonomer Betriebssystem-Prozess  Java Programmierschnittstelle und Grafikbibliothek zur Erstellung grafischer Oberflächen |
| Virtual-Maschine |  | Emulierte Umgebung für JAVA Applikationen |

# Konzept und Rahmenbedingungen

## Zeitrahmen

Das Gesamtprojekt soll über mehrere Schuljahre hinweg vom jeweiligen Jahrgang der 11. Klasse bearbeitet werden. Die Projektphase jedes einzelnen Schuljahres kann als eine Iteration betrachtet werden. Die Schüler bearbeiten (in Gruppen) pro Iteration ein fest umrissenes Teilaufgabengebiet. Der Projektleiter (=betreuende Lehrkraft) wählt eine der Lösungen zur Fortführung im folgenden Schuljahr aus. Besonders gute Teillösungen aus den verworfenen Projekten können im folgenden Schuljahr nachträglich integriert werden.

## Ziele des Projekts

Folgende Funktionspakete sind geplant (höchste Priorität = 1).

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Prio** | **Beschreibung** |
| 1 | 1 | Import folgender Stammdaten aus CSV-Dateien: Klassen, Schüler, Lehrer und deren persistente Speicherung in der Notenverwaltung. |
| 2 | 1 | Erfassung und persistente Speicherung von Unterrichts- und Zeugnisfächern. Ein Zeugnisfach kann aus mehreren Unterrichtsfächern bestehen. Zuordnung von Lehrern und Stundenanzahl zu den Unterrichtsfächern. |
| 3 | 1 | Erfassung von einzelnen Noten (eines Schülers) und deren sichere Speicherung. Dazu zählt auch die Definition von Leistungsarten und Leistungsgruppen. |
| 4 | 2 | Sammelerfassung von Noten eines Faches für eine komplette Klasse. Optional auch als Import von einer Excel-Datei oder per Zwischenablage (copy & paste). |
| 5 | 1 | Berechnung eines Vorschlags für die Halbjahres- und Jahresnoten unter Berücksichtigung von Gewichtungen (wie z. B. schriftlich zu mündlich und Unterrichtsfächern). |
| 6 | 3 | Erfassung einer (Halb-) Jahresnote, welche ggf. vom berechneten Vorschlag abweicht, d.h. die von der Software berechnete Note muss ggf. manuell überschrieben werden können. |
| 7 | 3 | Erstellung von druckbaren Übersichten (Notenblatt), die auch an den Schüler ausgehändigt werden können. |
| 8 | 4 | Übersicht aller Notenblätter eines Schülers während seiner Ausbildung an den EDV-Schulen (auch bei Wiederholung von Jahrgangsstufen) |
| 9 | 3 | Übernahme von Noten aus abgelegten Fächern ins Abschlusszeugnis |
| 10 | 4 | Nutzung des Programms in der Notenkonferenz: Übersicht der Noten eines Schülers; ggf. Durchschnittsnote der Vorrückungsfächer. |
| 11 | 4 | Druck von Zeugnissen (einschließlich Fehlzeiten und Bemerkungen). |
| 12 | 3 | Ein Berechtigungskonzept soll sowohl den Zugang zum Programm regeln (login), als auch die dem Benutzer zur Verfügung stehenden Funktionen einschränken. |
| 13 | 4 | Protokollierung aller Änderungen von Noten. |

Im Wesentlichen sind folgende Nutzungsziele geplant:

Ein anonymisiertes Muster eines Schülernotenblatts findet sich im Anhang dieses Dokuments.

## Benutzer / Zielgruppe

Die zu erstellende SW soll zunächst nur an den EDV-Schulen in Wiesau eingesetzt werden. Generell ist eine Nutzung an anderen Schulen in Betracht zu ziehen.

Die Hauptanwender des Produkts sind Lehrer, Klassenleiter und Verwaltungskräfte der EDV-Schulen Wiesau.

## Systemvoraussetzungen

Standard Windows PC, Java, ggf. MySQL-Datenbank, JasperReports (iReport)

## Einbindung in das Gesamtsystem

Es handelt sich um ein komplett neu zu erstellendes Programm. Allerdings soll die Erfassung redundanter Daten vermieden werden. Informationen über Schüler, Klassen und Fächer liegen im Digikabu vor und sollen von dort importiert bzw. abgeglichen werden können.

Aus Sicherheitsgründen ist von einem Betrieb der SW in einem abgegrenzten Netzwerk auszugehen.

# Beschreibung der Anforderungen

## Funktionale Anforderungen

### FP1 - Stammdaten Import

Folgende Stammdaten sind zu verwalten:

* Lehrer (Name, Vorname, Kürzel, Geburtsdatum, Geschlecht, Amtsbezeichnung, E-Mail)
* Schüler (Name, Vorname, Geburtsdatum, Konfession, Geschlecht)
* Klassen (Bezeichnung, Schuljahr, Schule)
* Fächer (Bezeichnung, Stundenmaß, Lehrer, unterteilt in Unterrichtsfächer (ja/nein) )
* Unterrichtsfächer (Bezeichnung, Stundenmaß, Lehrer)
* Schulen (z. B. Berufsfachschule, Fachschule etc.)

Aus dem Digikabu sollen die Unterrichtsfächer, Klassen und Schüler zu Beginn eines Schuljahrs übernommen werden. Schüleraustritt, Lehrerwechsel usw. muss auch während des laufenden Jahres übernommen werden können.

Ggf. müssen die Schülerdaten aus der Schulverwaltung »Atlantis« ergänzt werden.

### FP2 – Erfassung von Unterrichts- Zeugnisfächern

Abbildung des an den EDV-Schulen üblichen Systems von Zeugnisfächern und Unterrichtsfächern, d.h. mehrere Lehrer unterrichten ein Unterrichtsfach, welches mit entsprechender Gewichtung in die Gesamtnote des Faches einfließt. Beispielsweise setzt sich das Fach "Anwendungsprogrammierung" in der BFI11a aus den Unterrichtsfächern "Java" und "DBA" zusammen.

* Erfassung von Unterrichts- und Zeugnisfächern
* Zuordnung von Unterrichtsfächern zu Zeugnisfächern
* Zuordnung von Lehrern zu Unterrichtsfächern, inkl. Stundenmaß
* Vorgabe der Reihenfolge der Unterrichts- und Zeugnisfächer (maßgeblich für Darstellung auf dem Bildschirm, sowie für eine mögliche spätere Druckausgabe.

### FP3 – Erfassung von einzelnen Noten

##### Leistungsarten und Leistungsgruppen

Es sind verschiedene Leistungsarten und deren Gewichtung zu hinterlegen wie z. B. Schulaufgabe, Kurzarbeit, Abfragen, Stegreifaufgaben, Praktische Arbeit, Referat, Unterrichtsbeiträge.

Generell ist jede Leistungsart entweder der Leistungsgruppe der *mündlichen* oder der *schriftlichen* Noten zuzuordnen. Zur Gruppe der *mündlichen* Noten gehören die mündliche Abfrage, die Stegreifaufgabe, das Referat und der Unterrichtsbeitrag. Zur Gruppe der schriftlichen Noten gehören die Schulaufgabe, die Kurzarbeit und die Praktische Arbeit. Die Gewichtung von mündlicher Leistungsgruppe zu schriftlicher Leistungsgruppe ist je Schüler einstellbar(!) festzulegen.   
Beispiel: Gewicht schriftliche Leistungsgruppe = 2, Gewicht mündliche Leistungsgruppe = 1.

**Offener Punkt:** sollen die einzelnen Leistungsarten eine Gewichtung erhalten (z. B. Kurzarbeit = ½ Schulaufgabe?)

Lösungsvorschlag: Ja. Man beachte, dass sich die Gewichtung auf die einzelne Leistungsgruppe bezieht.  
Beispiel: Für die Gruppe der mündlichen Leistungserhebungen haben alle Leistungsarten das gleiche Gewicht 1. Dieses Gewicht bezieht sich nur auf die Leistungsarten der mündlichen Leistungserhebungen. Für die Gruppe der schriftlichen Leistungserhebungen haben Schulaufgabe und Praktische Arbeit das Gewicht 1 und die Kurzarbeit das Gewicht 0,5. Diese Gewichte beziehen sich nur auf die Leistungsarten der schriftlichen Leistungserhebungen.

##### Erfassung der Note

Macht die Auswahl einer Klasse, eines Schüler und eines (Unterrichts-)Faches, sowie einer Leistungsart notwendig.

In jedem der beschriebenen Fälle ist das Datum der Leistungserhebung (z. B. Schulaufgabe gehalten am 29.06.2016) zu erfassen.

**Offener Punkt:** was ist, wenn eine Schulaufgabe mehrere Unterrichtsfächer abdeckt?

Lösungsvorschlag: Die Note wird für jedes Unterrichtsfach eingetragen (natürlich mit entsprechender Gewichtung). (Diskussion notwendig!)

### FP4 – Sammelerfassung von Noten

macht die Auswahl einer Klasse und eines (Unterrichts-)Faches sowie einer Leistungsart notwendig. Copy & Paste aus dem Programm Excel wäre sehr wünschenswert.

### Protokollierung und Änderungshistorie

Die Eingabe, Änderung und Löschung von Noten soll in einer Änderungshistorie protokolliert werden (Datum, Uhrzeit, User, Note\_vorher, Note\_nachher). Wurde noch keine Note erfasst, so ist als Note\_vorher ein zu definierender Default-Wert (z. B. 0 oder -1) zu hinterlegen.

### Berechnung eines Vorschlags für Zeugnisnoten

Die Halbjahres- und Jahresnoten sollen pro Fach und Schüler anhand der Gewichtung der Leistungsarten, der Gewichtung der Unterrichtsfächer und der Gewichtung der Leistungsgruppen berechnet werden.

Beispiel:

Berechnung der schriftlichen Note:   
Im Fach "Anwendungsprogrammierung" (BFI11A) erhalten entsprechend der Anzahl der korrespondierenden Wochenstunden die Noten der schriftlichen Leistungsgruppe im Unterrichtsfach "Java" das Gewicht 7/9 und die Noten der schriftlichen Leistungsgruppe im Unterrichtsfach "DBA" das Gewicht 2/9.

Berechnung der mündlichen Note:  
Im Fach "Anwendungsprogrammierung" (BFI11A) erhalten alle mündlichen Leistungen das gleiche Gewicht.

Berechnung der Zeugnisnote:  
Entsprechend dem Gewicht der Leistungsgruppen wird aus der berechneten schriftlichen Note und der berechneten mündlichen Note ein Vorschlag für die Zeugnisnote berechnet.

Diese Vorschläge sollen aber manuell bearbeitet werden können, insbesondere dann, wenn die Note nahe „Komma fünf“ ist. Es soll im Programm ersichtlich sein, ob die automatisch berechnete Note verwendet wurde oder sie manuell festgelegt wurde.

### Login

Ein Login-Mechanismus für das Programm wäre wünschenswert.

Hier soll aber unbedingt auf bestehende und etablierte Mechanismen der EDV-Schulen zurückgegriffen werden (z. B. RADIUS-Server).

**Realisierungshinweis:** der RADIUS-Server ist auch aus dem Verwaltungsnetz erreichbar.

### Berechtigungskonzept

Im Programm soll ein Berechtigungskonzept vorgesehen werden. Dadurch soll sichergestellt werden, dass nur berechtigte Lehrer bzw. User Noten einsehen/ändern dürfen. Es sollen folgende Rollen und Berechtigungen unterschieden werden:

* Lehrer, der nicht in der Klasse unterrichtet:
  + keine Berechtigung.
* Lehrer der Klasse (Teilungen beachten, ggf. auch Lehrerwechsel zum Halbjahr):
  + Einsehen von sämtlichen Noten der Klasse
  + Bearbeitung der eigenen Noten
  + Bearbeitung der eigenen Zeugnisnoten
* Klassenleiter / stellvertretender Klassenleiter:
  + Einsehen von sämtlichen Noten der Klasse
  + Bearbeitung der eigenen Noten
  + Bearbeitung aller Zeugnisnoten
* Administrator:
  + Einsehen / Bearbeiten von sämtlichen Noten / Zeugnisnoten aller Klassen

Noten sollen nur für eine gewisse Zeit änderbar sein. Spätestens wenn das Zeugnis ausgehändigt ist, müssen diese gesperrt werden.

### Fehlzeiten [OPTIONAL]

Die Fehlzeiten eines Schülers sollen aus dem Digikabu importiert werden können:

Fehlend ganztags / davon unentschuldigt

Fehlend stundenweise / davon unentschuldigt

### Zeugnisbemerkungen [OPTIONAL]

### Druckausgaben

Die Fächer müssen sich für die Druckausgabe manuell sortieren lassen (z. B. oberstes Fach ist immer Religion)

Folgende Druckausgaben sind vorzusehen:

* Notenblatt als Übersicht (soll auch an Schüler ausgehändigt werden können)
* Zeugnis [OPTIONALE ANFORDERUNG]

**Realisierungshinweis**: es wäre wünschenswert, das bestehende Notenblatt, welches als JasperReport existiert, in den Notenmanager einzubinden.

### GUI

## Nicht-funktionale Anforderungen

### Aufbau

Das Programm soll in Schichten mit sinnvollen Schnittstellen aufgebaut sein.

### Fehlerkonzept

Das Programm soll ein einheitliches Fehlerkonzept realisieren. Sollte ein schwerwiegender Fehler auftreten, muss dieser dem Benutzer in einem Pop-up Fenster zur Anzeige gebracht werden; danach ist das Programm sauber zu beenden.

### Brauchbarkeit / Benutzerfreundlichkeit

Das Programm soll sich einfach bedienen lassen, konform zu den üblichen Konventionen von Windows-Programmen. Es muss auf einem beliebigen Windows-PC lauffähig sein.

### Zuverlässigkeit/Datensicherheit

Die Konsistenz der Datenbank ist stets zu gewährleisten. Personenbezogene Daten müssen vertraulich behandelt werden und vor unbefugtem Zugriff geschützt werden. Ein Backup-Mechanismus wäre wünschenswert, ist aber nicht zwangsweise Teil des Projekts.

### Performance

Das Programm muss sich flüssig bedienen lassen. Längere Wartezeiten sind zu vermeiden.

### Wartbarkeit / Erweiterbarkeit

Das Programm soll für künftige Änderungen oder Erweiterung vorbereitet sein. Dies ist durch eine sinnvolle Gliederung des Source-Codes in *packages* und Klassen zu realisieren. Ferner ist der Code ausreichend zu dokumentieren

Der Source Code ist mit der Versionskontrolle Subversion zu verwalten.

### Dokumentation

Es ist eine technische Dokumentation zu erstellen. Vorzugsweise enthält diese neben Erläuterungen in Textform auch Schaubilder (UML-Diagramme, ER‑Diagramm etc.)

# Gruppeneinteilung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Gruppe 1** | **Gruppe 2** | **Gruppe 3** | **Gruppe 4** | **Gruppe 5** |
| Prog. Sprache | Java | C# | Java | C# |  |
| PL | Kromer | Bechthold | Wunderlich | Ducheck |  |
| Stv. PL | Magerl | Tolnai | Stör | Köhler | Offen: |
|  | Vetter | Hasaltuntas | Gerstner | Bierler |  |
|  | Meister | Scholz | Weich | Stich |  |
|  | Kahl | Harrer | Bauer | Sammet | Kraus |
|  | Trettenbach?? | Petterson?? | Breitsprecher | Czech |  |
|  |  | Baylan?? | Sippel?? |  |  |

Folgende Gruppeneinteilung wurde im Rahmen des Unterrichts vorgenommen. Als Programmiersprachen, in denen die SW NotenManager realisiert wird, wurden Java und C# festgelegt.

# Risikobetrachtung

n.a.

# Abnahmekriterien / Testkonzept

Für die Entwicklung werden entsprechende Testdaten zur Verfügung gestellt. Die Abnahme des Produkts erfolgt im Rahmen des Programmierunterrichts.

# Offene Punkte

* Siehe Text bzw. jeweiliges Kapitel.

# Hilfreiche Tipps und Hinweise

Wie kann ich eine JTable nach verschiedenen Spalten sortieren?

[http://java.sun.com/docs/books/tutorial/uiswing/components/table.html#sorting](http://java.sun.com/docs/books/tutorial/uiswing/components/table.html%23sorting)

Erstellung von Reports mit Java:

<http://jasperforge.org/projects/jasperreports>

Anhang A

Gescanntes Schülernotenblatt (anonymisiertes Muster)

