

Dozentenbericht der ILV- Stundenplanung

IPA Jon Brunke

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Abbildungsverzeichnis.....	3
Tabellenverzeichnis.....	4
1 Analyse	5
1.1 Auftragsdivision.....	5
1.1.1 Erstellen des GUI	5
1.1.2 Entwicklung der Datenabfrage	5
1.1.3 Entwicklung der Datendarstellung	5
1.1.4 Testen und ausbessern der Lösung	5
1.2 Datenanalyse	6
1.2.1 Eigenschaft der Testdaten.....	6
1.2.2 Datenbank.....	6
1.2.3 Erforderliche Daten und deren Verfügbarkeit und Herkunft	8
1.2.4 Stundenumrechnung	10
1.3 Darstellung.....	10
2 Design.....	11
2.1 Use-Case	11
2.2 Erstellen des GUI	11
2.2.1 Ist-Zustand	11
2.2.2 Soll-Zustand.....	11
2.2.2.1 Skizze	12
2.3 Daten	13
2.3.1 Anpassungen an Testdaten.....	13
2.3.2 Inkonsistenz der Testdaten	13
2.3.3 Datentypen.....	13
2.3.3.1 Nicht Verfügbare Daten.....	13
2.3.3.2 Teilweise verfügbare Daten.....	13
2.3.3.3 Festgelegte Datenwerte	13
2.3.3.4 Berechnete Werte	13
2.3.3.5 Zusammengesetzte Werte	14
2.3.3.6 Datenbankdaten.....	14
2.3.4 Datenabfragen	14
2.3.4.1 Welche Dozenten.....	14
2.3.4.2 Personendaten.....	15
2.3.4.3 Stundendaten.....	15

2.3.4.4	Studententale	16
2.3.4.5	Saldodaten	16
2.4	Datendarstellung	16
2.4.1	Tabellen	16
2.4.2	Fusszeile	17
2.5	Klassen	17
2.5.1	Ist – Zustand	17
2.5.2	Soll – Zustand	18
2.5.2.1	Klassendiagramm	18
2.5.2.2	Beschreibung	18
2.5.3	Chronologischer Ablauf	20
3	Realisierung	21
3.1	Aufruf der Generierung	21
3.2	Abruf der benötigten Personen	21
3.3	Abrufen der Personen-, Stunden und Saldo-Daten	21
3.4	Darstellung der Daten	22
3.4.1	Tabellen	22
3.4.2	Fusszeile	22
3.4.3	Dateiname	23
3.5	Geleisteter Code	23
3.6	Code-Kommentierung	23
4	Testing	23
4.1	Testmethode	23
4.2	Testen während der Realisierung	23
4.3	Testfälle	24
4.4	Testbericht	25
4.5	Testauswertung	25
Anhang	25

Abbildungsverzeichnis

Abbildung 1 Datenbankausschnitt	6
Abbildung 2 Beispiel Detailblatt	8
Abbildung 3 Analyse Datendarstellung	10
Abbildung 4 UseCase	11
Abbildung 5 Start-Dialog	11

Abbildung 6 Benutzerschnittstelle - Dozentenberichte-Dialog.....	12
Abbildung 7 Personendaten - Leere Zellen.....	15
Abbildung 8 Stundendaten - Kategorie E.....	16
Abbildung 9 Datendarstellung - Tabelleneinteilung.....	17
Abbildung 10 Datendarstellung – Koordinatensystem.....	17
Abbildung 11 Klassendiagramm.....	18
Abbildung 12 Chronologischer Ablauf.....	21

Tabellenverzeichnis

Tabelle 1 Datenbank – Tables.....	7
Tabelle 2 Erforderliche Daten	10
Tabelle 3 Erforderliche Daten – Legende	10
Tabelle 4 Testing - Testfälle	24
Tabelle 5 Testing - Testbericht	25

1 Analyse

In der Analyse wird untersucht, welche Funktionen das Soll-Produkt können soll. Ausserdem können hier schon gewisse Schritte unternommen werden, um den Ablauf der Entwicklung festzulegen.

1.1 Auftragsdivision

Der Ablauf des Projekts kann in folgende Teilaufgaben unterteilt werden:

1.1.1 Erstellen des GUI

Bei dieser Teilaufgabe handelt es sich um die Erstellung des GUI für den Benutzer.

1.1.2 Entwicklung der Datenabfrage

Diese Teilaufgabe umfasst das Auslesen der Daten aus der Datenbank in eine geeignete Form im Code. Ausserdem die Erstellung der Daten, welche nicht aus der Datenbank ausgelesen werden können, wie etwa berechnete oder fehlende Daten oder leere Zellen und deren hinzufügen zu den ausgelesenen Daten. Bei der Speicherung eben dieser Daten muss darauf geachtet werden, dass alle relevanten Daten ausgelesen und so aufbereitet, dass sie in der nächsten Aufgabe ohne Probleme verwendet werden können. Diese Aufgabe lässt sich weiter in die folgenden Schritte aufteilen:

- Personaldaten
- Stundendaten
- Studententale
- Saldodaten

1.1.3 Entwicklung der Datendarstellung

Dieser Schritt beinhaltet das Darstellen der Daten in geeigneter Form, also so, dass das Ergebnis der Excel-Vorlage möglichst ähnlich sieht. Auch hier ist eine weitere Unterteilung möglich. Nämlich in die Schritte:

- Personaldaten
- Stundendaten
- Studententale
- Saldodaten
- Fusszeilen

1.1.4 Testen und ausbessern der Lösung

Der letzte Schritt dreht sich darum, die entwickelte Lösung zu testen und allfällige Mängel auszubessern. Dieser Schritt ist sehr wichtig und kann unter Umständen viel Zeit beanspruchen. Diese Aufgabe kann ebenfalls in weitere Schritte aufgeteilt werden:

- Testfälle erstellen
- Testen
- Testen auswerten
- Korrekturen vornehmen

1.2 Datenanalyse

Hier wird untersucht, welche Daten für das Projekt von Nöten sind und woher diese genau kommen. Dies erleichtert im späteren Verlauf das Erstellen der Datenabfrage.

1.2.1 Eigenschaft der Testdaten

In der Datenbank sind nur Testdaten enthalten, um die Stundenerfassung der Kostenstelle „TIMES“ für das Studienjahr 13/14 abzubilden.

1.2.2 Datenbank

Diese Abbildung zeigt den für diese Arbeit wichtigen Teil der Datenbank, sowie deren Beziehungen:

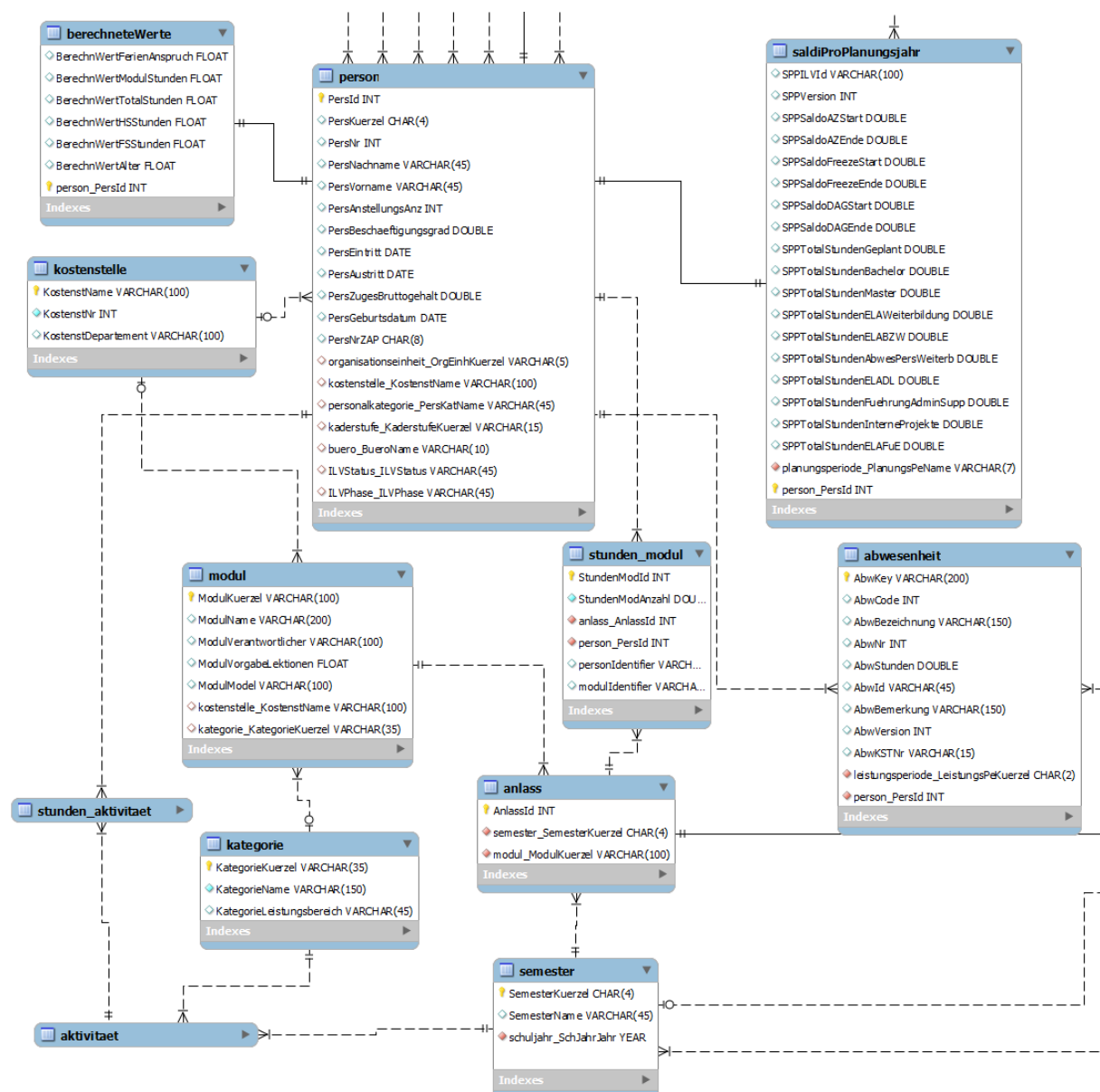


Abbildung 1 Datenbankausschnitt

Um die Datenbank noch ein wenig genauer zu erläutern sind hier die wichtigsten Tabellen mit den Primary Keys und einer kurzen Beschreibung der enthaltenen Daten aufgelistet:

Table	Primary Key	Beschreibung
person	PersId	Dieser Table enthält Daten, welche allgemeine Informationen wie Name, Nachname, Peronal-Nr., etc. beinhalten.
stunden_modul	StundenModId	Enthält die Anzahl Stunden welche eine Person in einem bestimmten Anlass leistet.
anlass	AnlassId	Enthält Informationen zu Anlässen, also das Vorhandensein eines Moduls in einem Semester.
semester	SemesterKuerzel	Enthält Informationen zu den Semestern.
modul	ModulKuerzel	Enthält Informationen zu den Modulen wie etwa Name, Kostenstelle, Kategorie, etc.
kostenstelle	KostenstName	Hier sind Informationen zu Kostenstellen enthalten. Z.B. Name und Nummer.
saldiProPlanungsjahr	person_PersId	Hier sind Daten wie das Vorjahres- und andere Saldi enthalten.
berechneteWerte	person_PersId	Dieser Table enthält Daten wie z.B. das Alter oder den Ferienanspruch enthalten.

Tabelle 1 Datenbank – Tables

Jegliche Id-Werte, also zum Beispiel PersId oder AnlassId, sind künstliche Schlüssel welche erstellt wurden, da die restlichen Daten nicht sehr konstant sind. Jegliche Abfragen aus den betreffenden tables werden über diese Schlüssel getätigt.

1.2.3 Erforderliche Daten und deren Verfügbarkeit und Herkunft

Als Vorlage ist ein Detailblatt in Form eines Excel-Sheets vorhanden, woraus die abzubildenden Daten entnommen werden können. Das Detailblatt ist hier abgebildet:

Leistungsplanung : [Studienjahr] - 1. August [S bis 31. Juli [S]						
Name/Vorname		Personal-Nr.		BG-V	0%	Personendaten
Geburtsdatum		Kurzzeichen		BG-L	0%	
Kostenstelle		Ferienpauschale in Stunden (bez. BG-L)			210.00	
Eintritt		Weiterbildungspauschale in Stunden (bez. BG-L)			168.00	
Bruttoarbeitszeit	2088.60 (bei BG 100%)	Bruttoarbeitszeit (bez. BG-L)		Total Stunden	2088.60	
Saldo-Vorjahr	295.04	Nettoarbeitszeit (bez. BG-L)		Total Stunden	1945.83	

Bezeichnung	HS h	%	FS h	%	SJ h	Stundendaten
A - Bachelor: Lehre						
Semester HS[SemesterHS] :						
MFEM (3) /	110	6			110	
MFEM (3 + 3) /	110	6			110	
BME1 /	55	3			55	
Semester FS[SemesterFS] :						
MFL2-P FEM (2 Grp, 0.38 SWL) /			14	1	14	
LT2 /			74	4	74	
B - Bachelor: Weitere Leistungen						
Semester HS[SemesterHS] :						
Physio Hnln /	57	3			57	
PA andere Studiengänge /	80	4			80	
Semester FS[SemesterFS] :						
TEMS /			147	8	147	
C - Projekt - und Bachelor-Arbeiten						
Projektarbeit(en)	120	6			120	
Bachelorarbeit(en)			100	5	100	
D - Master MSE						
Semester HS[SemesterHS] :						
MSE Unterricht	2	0			2	
Semester FS[SemesterFS] :						
MSE Advisor			36	2	36	
MT Cyril Huber			56	3	56	
MT Nicolas Elkuch			56	3	56	
E1 - ELA: Weiterbildung						
E2 - ELA: Forschung & Entwicklung						
Übertrag aus Planung	430	22	250	13	680	
E3 - ELA: Dienstleistungen						
E4 - Interne Projekte/Sonderprojekte						
E5 - Führung/Administration/Support						
Übertrag aus Planung	25	1	223	11	248	
F - Abwesenheiten und persönliche Weiterbildung						
Ferien	105		105		210	
pers. Weiterbildung (Pauschale)	84		84		168	
Total	1179	51	1145	49	2324	
prozentuale Auslastung bezogen auf Nettoarbeitszeit						
1946 h	Lehre	52%	ELA	35%	Diverses	
100%					13%	

Bemerkungen zu Tätigkeiten:		Besprochen am	Saldodaten
Voraussichtlicher Saldo per 31.07.2013		295.04	
Sollarbeitszeit SJ 13/14	2088.60		
Total Stunden geplant	2323.83		
Differenz geplant	235.23	235.23	
Voraussichtlicher Saldo per 31.07.2014		530.27	

Abbildung 2 Beispiel Detailblatt

Da es darum geht dieses Detailblatt möglichst gut abbilden zu können, werden hier die darauf enthaltenen Daten, sowie deren Herkunft und Verfügbarkeit aufgelistet:

Daten	Herkunft	Verfügbarkeit (in DB)
Personendaten		
Name	Datenbank (person, PersNachname)	
Vorname	Datenbank (person, PersVorname)	
Geburtsdatum	Datenbank (person, PersGeburtsdatum)	
Kostenstelle-Name	Datenbank (person, kostenstelle:KonstenstName)	
Kostenstelle-Nr.	Datenbank (kostenstelle, KostenstNr)	
Eintritt	Datenbank (person, PersEintritt)	
Personal-Nr.	Datenbank (person, PersNr)	
Kurzzeichen	Datenbank (person, PersKuerzel)	
Bruttoarbeitszeit	Variable im Code	-
Saldo-Vorjahr	Datenbank (saldiproplanungsjahr, SPPSaldoAZStart)	
BG-V (Beschäftigungsgrad Vertrag)	Datenbank (person, PersBeschaeftigungsgrad)	
BG-L (Beschäftigungsgrad Lohn)	Datenbank (person, PersZugesBruttogehalt)	
Ferienpauschale in Stunden (bez. BG-L)	Datenbank (abwesenheit, AbwStunden)	
Weiterbildungspauschale in Stunden (bez. BG-L)	Datenbank (abwesenheit, AbwStunden)	
Bruttoarbeitszeit (bez. BG-L)	Berechnet	-
Nettoarbeitszeit (bez. BG-L)	Berechnet	-
Datenbezeichnungen	Variablen	-
Stundendaten		
Modulkategorien	Variablen	-
Module im jeweiligen Semester	Datenbank (modul/anlass)	
Stunden in den Modulen	Datenbank (modul/anlass/stunden_modul)	
Ferien (pro Semester)	Berechnet	-
Pers. Weiterbildung(pro Semester)	Berechnet	-
Ferien (Total Schuljahr)	Berechnet	-
Pers. Weiterbildung (Total Schuljahr)	Berechnet	-
Prozentualer Anteil an Nettoarbeitszeit pro Modul und Stunden	Berechnet	-
Totale der Stunden für die Semester (Stundenanzahl)	Datenbank (anlass/stunden_modul)	
Total der Stunden für das Schuljahr	Datenbank (anlass/stunden_modul) / berechnet	
Total der Stunden für die Semester (Prozente)	Berechnet	-
Saldodaten		
Voraussichtliches Saldo per [Vorgangsjahr]	= Saldo-Vorjahr	Siehe Saldo-Vorjahr
Sollarbeitszeit [aktuelles Schuljahr]	= Bruttoarbeitszeit (bez. BG-L)	Siehe Bruttoarbeitszeit (bez. BG-L)
Total Stunden geplant	= Total der Stunden für das Schuljahr	Siehe Total der Stunden für das Schuljahr
Differenz geplant	Berechnet	-

Voraussichtlicher Saldo per [aktuelles Jahr]	Berechnet	-
Prozentuale Auslastung bezogen auf Nettoarbeitszeit	Berechnet	-
Bemerkungen zu Tätigkeiten	Variable / = Kurzzeichen/Nachname, Vorname	Siehe Kurzzeichen, Name, Vorname
Titel, Fussnote. Etc.		
Titel	Variable	-
Dateiname	Variable / =Kurzzeichen/Nachname, Vorname	-
Aktuelles Datum	Variable	-

Tabelle 2 Erforderliche Daten

Die folgende Legende erläutert die Farbkodierung der Verfügbarkeit der Daten:

Farbe	Beschreibung	Bemerkung
	Die Daten sind vorhanden	Diese Daten sind für jeden benötigten Dozenten vorhanden
	Die Daten sind teilweise vorhanden	Diese Daten sind nicht bei jedem benötigten Dozenten vorhanden
	Die Daten sind nicht vorhanden	Diese Daten sind für keinen benötigten Dozenten vorhanden

Tabelle 3 Erforderliche Daten – Legende

1.2.4 Stundenumrechnung

Bei einigen Modulen werden die Stunden in Schulwochenlektionen oder als Arbeiten festgehalten. Für diese Fälle existieren folgende Umrechnungswerte:

- Schulwochenlektionen
SWL*36.8
- BA-Arbeit
Anzahl*50
- PA-Arbeit
Anzahl*40

1.3 Darstellung

Um die Daten darzustellen wird ReportLab verwendet. Um die Daten möglichst originalgetreu abzubilden, entschloss ich mich die Daten in eine Kombination verschiedener Tabellen darzustellen. Hierbei ist es so, das ReportLab die Daten in die Tabelle aus einem multidimensionalen Array liest. Das bedeutet, dass auch Datenbezeichnungen in diesen Array eingetragen werden müssen. Hier ein Beispiel dazu:

data = [['Name/Vorname', '████████', '████████', 'Personal-Nr.', '████████', 'BG-V', '0%'], ['Geburtsdatum', '████████', '██', 'Kurzzeichen', '███', 'BG-L', '0%']]							
	data[0][0]	data[0][1]	data[0][2]	data[0][3]	data[0][4]	data[0][5]	data[0][6]
data[0]	Name/Vorname	████████	████████	Personal-Nr.	████████	BG-V	0%
data[1]	Geburtsdatum	████████		Kurzzeichen	███	BG-L	0%
	data[1][0]	data[1][1]	data[1][2]	data[1][3]	data[1][4]	data[1][5]	data[1][6]

Abbildung 3 Analyse Datendarstellung

2 Design

In diesem Kapitel geht es um das Designen des Lösungsansatzes.

2.1 Use-Case

Dieses Diagramm zeigt auf, welche Aktion der Benutzer ausführen kann. Dazu gehören das Erstellen aller Dozentenberichte der betreffenden Kostenstelle (in diesem Fall auf Grund der Testdaten nur „T IMES“) sowie das Erstellen eines einzelnen Stundenberichts auf Grund der Eingabe der Dozentendaten des gewünschten Dozenten.

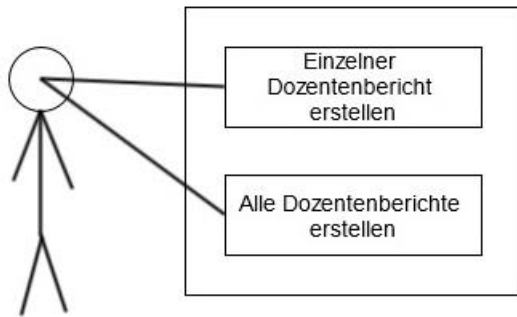


Abbildung 4 UseCase

2.2 Erstellen des GUI

Hier werden Überlegungen zur Benutzerschnittstelle festgehalten.

2.2.1 Ist-Zustand

Als Benutzerschnittstelle dient aktuell ein Python-GUI. Dabei kann der Benutzer anfangs auswählen, ob er die Dozentenberichte generieren, oder zur Stundenerfassung gelangen will. Wobei uns hier nur das Generieren der Dozentenberichte interessiert. Dieses ist hier abgebildet:

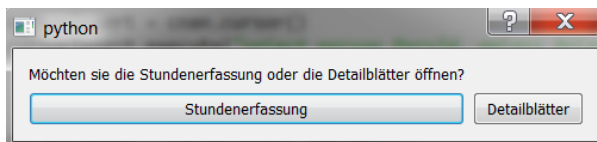


Abbildung 5 Start-Dialog

2.2.2 Soll-Zustand

Nach dem Betätigen des Buttons „Detailblätter“ des aktuellen GUIs soll diese geschlossen und ein weiteres Aufgerufen werden. Darauf sollen folgende Elemente und Funktionen enthalten beziehungsweise aufrufbar sein:

- Eingabe des Personenkürzels des Benutzers.
- Eingabe der betreffenden Kostenstelle (wird auf Grund der Testdaten standardmässig auf „T IMES“ gesetzt).
- Erstellen der Detailblätter aller Dozenten basierend auf einer Eingabe der betreffenden Kostenstelle.
- Eingabe des Personenkürzels, des Nachnamen und des Vornamen eines bestimmten Dozenten.
- Erstellen eines einzelnen Detailblattes basierend auf der Eingabe dessen Personenkürzels.

- Ein „Close“-Button welcher die Anwendung beendet.

2.2.2.1 Skizze

Diese Skizze spiegelt das Soll-Aussehen des GUI wieder. Enthalten sind folgende Elemente:

- Ihr Personenkürzel
Hier wird der Personenkürzel des Benutzers eingegeben
- Ihre Kostenstelle
Hier wird die Kostenstelle eingegeben, von welcher die Berichte erstellt werden sollen. Auf Grund der Eigenschaften der Testdaten ist dieser Wert Standardmässig auf „T IMES“ gesetzt.
- Gewünschter Dozent
Diese Eingaben bestimmen die Werte, welche zur Erstellung eines einzelnen Berichts notwendig sind. Auf Grund dessen, dass bei einigen Personen kein Personenkürzel oder keine Namensinformationen vorhanden sind, müssen hier der Kürzel, der Vorname sowie der Nachname eingegeben werden:
 - Personenkürzel
Personenkürzel des gewünschten Dozenten
 - Nachname
Nachname des gewünschten Dozenten
 - Vorname
Vorname des gewünschten Dozenten
 - Bericht generieren
Ruft das Erstellen des einzelnen Berichts auf Grund der Eingabe der oben genannten Werte auf.
- Alle Berichte generieren
Erstellt alle Berichte der eingegebenen Kostenstelle (in diesem Fall auf Grund der Testdaten nur „T IMES“)
- Close
Schliesst die Anwendung

Ihr Personenkürzel:

Ihre Kostenstelle:

Gewünschter Dozent:

Personenkürzel:

Nachname:

Vorname:

Abbildung 6 Benutzerschnittstelle - Dozentenberichte-Dialog

2.3 Daten

2.3.1 Anpassungen an Testdaten

In der Datenbank sind nur Testdaten enthalten, welche die Stundenerfassung der Kostenstelle „T IMES“ für das Studienjahr 13/14 wiedergeben. Deshalb können nur Berichte für diese Kostenstelle erstellt werden, was bedeutet das einige Variablen auf diesen Wert festgesetzt werden müssen.

2.3.2 Inkonsistenz der Testdaten

Es sind nicht bei allen Personen alle Personendaten in der Datenbank vorhanden. Sollte bei einer Person ein oder mehrere Werte nicht vorhanden sein, wird der Wert auf „nA“ gesetzt.

2.3.3 Datentypen

Hier werden die in den erforderlichen Daten festgehaltenen Datenherkünfte genauer erläutert.

2.3.3.1 Nicht Verfügbare Daten

Gewisse Daten sind in der Datenbank nicht vorhanden. Um diese Daten trotzdem abbilden zu können werden sie im Code als Variable festgesetzt.

- Ferienpauschale in Stunden (bez. BG-L) = 210.00
- Weiterbildungspauschale in Stunden (bez. BG-L) = 168.00
- Ferien (pro Semester) = 105.00
- Pers. Weiterbildung (pro Semester) = 84.00
- Ferien (Total Schuljahr) = 210.00
- Pers. Weiterbildung (Total Schuljahr) = 168.00

2.3.3.2 Teilweise verfügbare Daten

Teilweise vorhandene Daten werden wenn sie nicht vorhanden sind im Code auf feste Werte gesetzt. Bei Werten welche keine Zahlen sind ist dieser Wert meistens „nA“ bei Zahlenwerten „0.00“

2.3.3.3 Festgelegte Datenwerte

Einige Daten werden im Code als Variable mit einem festgelegten Wert versehen:

- Modulkategorien
Die Kategorien werden gemäss der Excel-Lösung benannt und in einer Liste festgehalten.
- Bruttoarbeitszeit
Wird wie in der Excel-Lösung mit dem Wert „2088.60“ versehen.
- Aktuelles Datum
Wird aus dem System abgefragt
- Titel
Der Titel wird auf „Leistungsplanung: 13/14 - 16. September 2013 - 28. Juni 2014“
- Datenbezeichnungen
Die verschiedenen Datenbezeichnungen wie z.B. „Name/Vorname“, „Personal-Nr.“, etc. werden als Variablen in den Array mit den entsprechenden Daten eingetragen.

2.3.3.4 Berechnete Werte

Folgende Daten werden mittels anderer Daten berechnet:

- Bruttoarbeitszeit (bez. BG-L)

(Bruttoarbeitszeit)*(BG-V rechnet).

- Nettoarbeitszeit (bez. BG-L)
Ist die Summe der im zu erfassenden Jahr aufgeführten Stunden ohne Ferien und pers. Weiterbildung.
- Prozentualer Anteil an Nettoarbeitszeit pro Modul und Stunden
Ist der prozentuale Anteil der in einem bestimmten Modul geleisteten Stunden an der Nettoarbeitszeit
- Totale der Stunden für die Semester (Prozente)
Prozentualer Anteil des Studentotals des betreffenden Semesters am Studentotal des Schuljahres
- Differenz geplant
(Sollarbeitszeit [aktuelles Schuljahr]) – (Total Stunden geplant)
- Voraussichtlicher Saldo per [aktuelles Jahr]
Voraussichtliches Saldo per [Vorgangsjahr] + (Differenz geplant)
- Prozentuale Auslastung bezogen auf Nettoarbeitszeit
Sind die prozentualen Anteile der einzelnen Modulkategorien an der Nettoarbeitszeit
- Total der Stunden für das Schuljahr
Nettoarbeitszeit + Ferien + pers. Weiterbildung

2.3.3.5 *Zusammengesetzte Werte*

Einige Werte setzen sich aus verschiedenen anderen Werten zusammen:

- Dateiname (Fusszeile)
Setzt sich aus einem definierten String und dem Personenkürzel des betreffenden Dozenten zusammen.
- Bemerkungen zu Tätigkeiten
Setzt sich aus einem definierten String, dem Personenkürzel des Benutzers sowie dem Personenkürzel des betreffenden Dozenten zusammen.

2.3.3.6 *Datenbankdaten*

Folgende Daten können aus der Datenbank mittels einem oder eine Kombination von Selects ausgelesen werden:

- | | |
|-------------------------------------|---|
| • Name | • BG-L (Beschäftigungsgrad Lohn) |
| • Vorname | • BG-V (Beschäftigungsgrad Vertrag) |
| • Geburtsdatum | • Module im jeweiligen Semester |
| • Kostenstelle | • Stunden in den Modulen |
| • Eintritt | • Totale der Stunden für die Semester (Stundenanzahl) |
| • Personal-Nr. | • Total der Stunden für das Schuljahr |
| • Kurzzeichen | |
| • Saldo-Vorjahr | |
| • BG-V (Beschäftigungsgrad Vertrag) | |

2.3.4 *Datenabfragen*

2.3.4.1 *Welche Dozenten*

Bei den Datenabfragen geht es als erstes darum festzulegen, von welchen Dozenten die Daten abgefragt werden sollen:

- Alle Personen, welche zur Kostenstelle „T IMES“ gehören.
- Alle Personen, welche in einem Modul mit der Kostenstelle „T IMES“ unterrichten.
- Alle Anlässe, welche sich auf ein Modul mit der Kostenstelle „T IMES“ beziehen.
- Alle Anlässe, welche sich auf eine Person mit der Kostenstelle „T IMES“ beziehen.

Anschliessend sind aus diesen Daten die Dozenten auszuschliessen, für welche keine Stundeneinträge vorhanden sind, also jene welche keine Stunden leisten.

2.3.4.2 Personendaten

Für jeden Dozenten wird eine separate Datenabfrage durchgeführt. Dabei werden in einem ersten Schritt alle Personendaten welche in der Datenbank vorhanden sind ausgelesen und in einen mehrdimensionalen Array gespeichert.

In einem zweiten Schritt wird ein weiteren mehrdimensionaler Array erstellt, welcher zum einen alle Daten des ersten Schrittes sowie alle noch benötigten Daten wie etwa Datenbezeichnungen, berechnete oder festgelegte Werte oder zur korrekten Darstellung benötigte leere Werte enthält.

Man könnte hier auch sofort alle Daten der Dozenten abfragen und anschliessend die Filterung vollziehen, dies hat jedoch bei einem kurzen Testversuch nicht zu einem qualitativ passenden Ergebnis geführt und wurde wieder verworfen.

Der Umstand, dass für die korrekte Darstellung leere Werte, also leere Zellen eingefügt werden müssen ist hier ersichtlich:

Leistungsplanung : [Studienjahr] - 1. August [S bis 31. Juli [S]							
Name/Vorname			Personal-Nr.			BG-V	0%
Geburtsdatum			Kurzzeichen			BG-L	0%
Kostenstelle		T IMES	Ferienpauschale in Stunden (bez. BG-L)				210.00
Eintritt			Weiterbildungspauschale in Stunden (bez. BG-L)				168.00
Bruttoarbeitszeit	2088.60	(bei BG 100%)	Bruttoarbeitszeit (bez. BG-L)		Total Stunden		2088.60
Saldo-Vorjahr	295.04		Nettoarbeitszeit (bez. BG-L)		Total Stunden		1945.83

Abbildung 7 Personendaten - Leere Zellen

Danach ist ein multidimensionaler Array vorhanden, welcher alle benötigten Personendaten in für die Darstellung korrekter Weise gespeicherter Art enthält.

2.3.4.3 Stundendaten

Auch bei den Stundendaten ist es so, dass für jeden Dozenten eine separate Abfrage ausgeführt wird.

In einem ersten Schritt werden hierbei die Module, in welchen der betreffende Dozent Stunden absolviert pro Kategorie ausgelesen und in einen mehrdimensionalen Array gelesen. Danach werden für jedes dieser Module die vom Dozenten in den beiden Semestern geleisteten Stunden ausgelesen und dem Array hinzugefügt. Hierbei werden auch die in der Analyse erwähnten Stundenumrechnungen gemacht.

In einem zweiten Schritt wird ein weiterer mehrdimensionaler Array gebildet, welche alle Daten des ersten Schrittes enthält und diese um noch nicht vorhandene benötigte Werte ergänzt. Dazu können Datenbezeichnungen oder berechnete, festgelegte oder leere Werte gehören.

Anschliessend ist für jede Kategorie eine mehrdimensionaler Array vorhanden, welcher die benötigten Daten in für die Darstellung idealer weise beinhaltet.

Hierbei ist festzuhalten das diese Sektion aus der Vorlage:

E1 - ELA: Weiterbildung					
E2 - ELA: Forschung & Entwicklung					
Übertrag aus Planung	430	22	250	13	680
E3 - ELA: Dienstleistungen					
E4 - Interne Projekte/Sonderprojekte					
E5 - Führung/Administration/Support					
Übertrag aus Planung	25	1	223	11	248

Abbildung 8 Stundendaten - Kategorie E

In eine einzelne Kategorie zusammengefügt wird.

Ausserdem werden die Daten der Kategorie F, auf Grund dessen, dass diese Daten nicht vorhanden sind auf Standardwerte gesetzt.

2.3.4.4 Studentotale

Auch bei den Stundendaten ist es so, dass diese für jeden Dozenten einzeln abgefragt werden. Dabei werden erst die aus der Datenbank benötigten Daten abgefragt.

Anschliessend werden, im Gegensatz zu der Abfrage der Personen- oder Stundendaten, berechnete oder festgelegte Werte dem Array schon im selben Schritt hinzugefügt. Auch die in der Analyse erwähnte Stundenumrechnung wird hier durchgeführt.

Danach ist ein mehrdimensionaler Array vorhanden, welcher alle Daten in für die Datendarstellung korrekter Weise gespeicherter Art enthält.

2.3.4.5 Saldodaten

Die Saldodaten werden ebenfalls für jeden Dozenten einzeln ausgelesen. Dabei werden wie bei den Studentotalen die Datenbankdaten, berechnete oder festgelegte Werte, etc. in einem Schritt in einen mehrdimensionalen Array gefügt.

2.4 Datendarstellung

2.4.1 Tabellen

Für die Datendarstellung werden die in den Datenabfragen erstellten Arrays der betreffenden Klasse übergeben.

Die Daten werden mittels verschiedener Tabellen dargestellt. Die Einteilung ist dabei wie folgt:

- Personendaten
- Stundendaten
 - Bachelormodule
 - Restl. Kategorien
 - Totale
- Saldodaten

Hier eine Verdeutlichung der Einteilung der Stundendaten in die Kategorien:

Bezeichnung		HS h	%	FS h	%	SJ h	
A - Bachelor: Lehre							BA A
Semester HS[SemesterHS] :							
MFEM (3) /		110	6			110	
MFEM (3 + 3) /		110	6			110	
BME1 /		55	3			55	
Semester FS[SemesterFS] :							
MFL2-P FEM (2 Grp, 0.38 SWL) /				14	1	14	
LT2 /				74	4	74	
B - Bachelor: Weitere Leistungen							BA B
Semester HS[SemesterHS] :							
Physio Hnln /		57	3			57	
PA andere Studiengänge /		80	4			80	
Semester FS[SemesterFS] :							
TEMS /				147	8	147	
C - Projekt - und Bachelor-Arbeiten							MA
Projektarbeit(en)		120	6			120	
Bachelorarbeit(en)				100	5	100	

Abbildung 9 Datendarstellung - Tabelleneinteilung

Die Formatierungskonfigurationen der einzelnen Tabellen können bei ReportLab ebenfalls in einem Array gespeichert werden. Hierbei sind verschiedene Funktionen wie *LINEABOVE*, *BACKGROUND* oder *GRID* verfügbar, welche jeweils auf einen bestimmten Zellenbereich angewendet werden können. Wobei die Zellen in einem bestimmten Koordinatensystem gekennzeichnet werden:

0,0				
	Dozent	Modul	Stunden	
	Doz1	Mod1	2	
	Doz2	Mod2	3	
	Doz3	Mod3	4	
	Doz4	Mod4	6	
				-1, -1

Abbildung 10 Datendarstellung – Koordinatensystem

Daher werden für die Darstellung Arrays mit der Formatierung für jeweils die Personendaten, die Stunden-und Studenttotal-Daten und die Saldodaten erstellt.

2.4.2 Fusszeile

Die Fusszeile wird als canvas auf jede Seite gezeichnet.

2.5 Klassen

2.5.1 Ist - Zustand

Zum Projektbeginn existierten folgende Klassen:

- StartDialog
- PlanungstoolWidget
- StundenWidget
- KategorieWidget
- ErfassungsWidget
- TotaleWidget
- ButtonWidget

Für dieses Projekt wird hierbei nur die Klasse StartDialog benötigt, welche um eine Funktion zum Aufruf des Dozentenberichte-Tool erweitert wurde.

2.5.2 Soll - Zustand

2.5.2.1 Klassendiagramm

In diesem Klassendiagramm sind die erstellten Klassen ersichtlich, sowie die Funktionen, welche jeweils eine Klasse aufrufen.

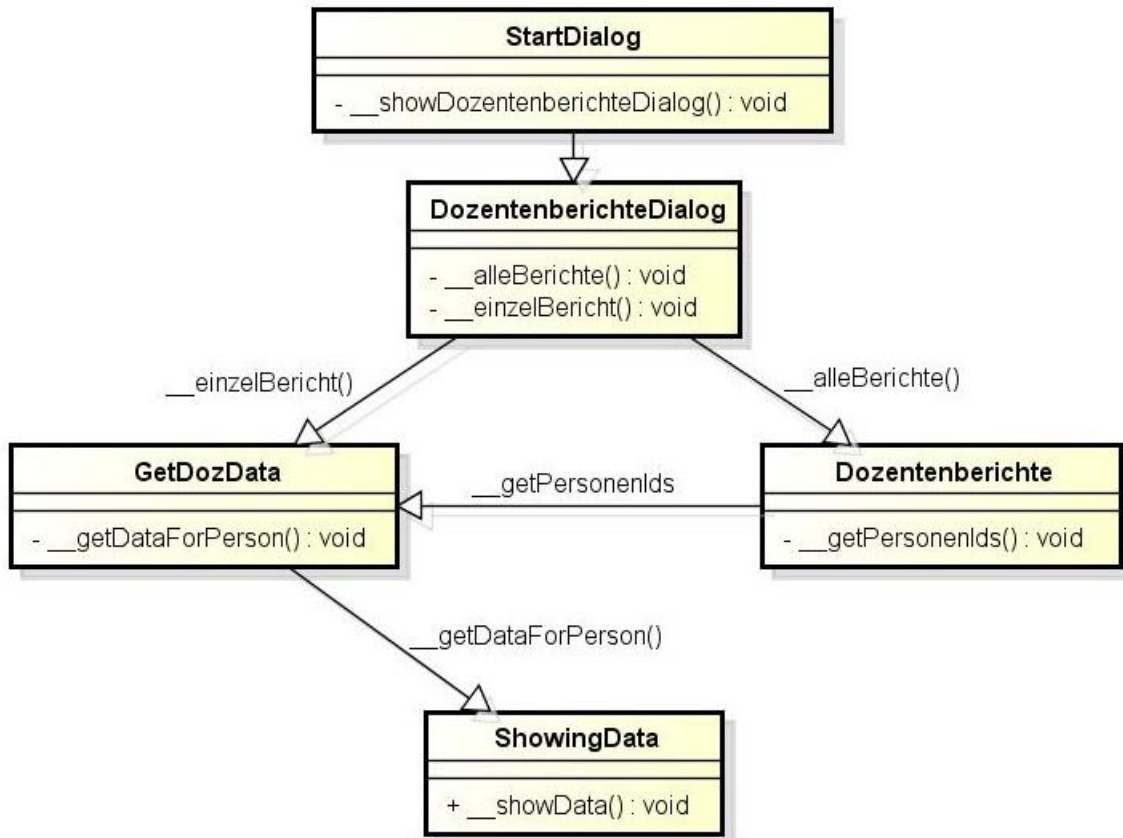


Abbildung 11 Klassendiagramm

2.5.2.2 Beschreibung

Um die Klassen noch ein wenig genauer zu beschreiben, werden sie hier aufgelistet und erläutert:

- **DozentenberichteDialog**
Diese Klasse wird vom StartDialog konstruiert und beinhaltet das GUI des Dozentenberichts-Tool, über welches die Klasse Dozentenberichte aufgerufen wird. Dies wird mittels folgenden Funktionen bewerkstelligt:
 - `__init__`
Ist der Konstruktor dieser Klasse
 - `setupUI`
Erstellt den eigentlichen Dialog und zeigt diesen an. Ruft nach entsprechender Eingabe die Funktionen `__close`, `__alleBerichte` oder `__einzelBericht` auf.
 - `__close`

- Schliesst den Dialog
- __alleBerichte
Konstruiert die Klasse Dozentenberichte, um alle benötigten Dozentenberichte zu erstellen und gibt ihr benötigte Werte mit: Personenkürzel des Benutzers. Vor dem Aufruf überprüft sie ausserdem das Vorhandensein dieser Werte.
- __einzelBericht
Konstruiert auf Grund der Eingabe bei des Benutzers die Klasse GetDozData zur Erstellung eines einzelnen Berichts und übergibt ihr die benötigten Werte: Ihr Personenkürzel, Personenkürzel des Benutzers, Nachname und Vorname des gewünschten Dozenten. Vor dem Aufruf überprüft sie ausserdem das Vorhandensein dieser Werte.
- __errorDialog
Falls benötigte Eingabewerte nicht vorhanden sind, wird diese Funktion aufgerufen, welche eine kleine Fehlermeldung anzeigt.
- Dozentenberichte
Hier werden die PersonenIds der benötigten Dozenten abgerufen. Anschliessend wird die Klasse GetDozData konstruiert, welcher ein Array mit den PersonenIds übergeben wird.
Folgende Funktionen sind hier enthalten:
 - getPersonenIds
Hier werden die benötigten PersonenIds abgerufen werden. Anschliessend wird die Klasse Dozentenberichte konstruiert, welcher diese PersonenIds übergeben werden.
- GetDozData
Diese Klasse beinhaltet den Code, welcher zur Datenabfrage und deren Speicherung zuständig ist. Sie konstruiert am Ende die Klasse ShowingData auf und übergibt ihr die abgerufenen Daten. Folgende Funktionen sind hier enthalten:
 - __getDataForPers
Diese Funktionen ruft für jeden Eintrag in dem von der Klasse Dozentenberichte erhaltenen Array die unten aufgelisteten Funktionen auf und übergibt Ihnen jeweils den aktuellen Eintrag, also die PersId. Anschliessend konstruiert sie die Klasse ShowingData auf und übergibt ihr die in den folgenden Funktionen erstellten Arrays.
 - __getStundenTotale
Ruft basierend auf der PersonenId die verschiedenen Studententotalen welche darzustellen sind ab und speichert diese.
 - __getPersonenData
Ruft basierend auf der erhaltenen PersId die Personendaten der betreffenden Person ab und speichert diese in einen Array.
 - __insertVariablesPersonenData
Erstellt einen Array, welcher die in __getPersonenData abgerufenen Werte enthält sowie noch nicht enthaltene Daten wie etwa Bezeichnungen, berechnete Werte oder zur Darstellung benötigte leere Einträge.
 - __getModuls
Lädt basierend auf der erhaltenen PersonenId die Module ab, in welchen der Dozent Stunden absolviert. Diese werden anschliessend in einen Array gespeichert.
 - __getModulStunden

- Ruft für jedes der im Array von `__getModuls` gespeicherten Module die Stunden des jeweiligen Moduls ab und fügt diese dem Array hinzu.
 - `__insertVariablesModule`
Erstellt einen Array welcher die Module mit ihren Stunden enthält, sowie noch nicht enthaltene Werte wie etwa Bezeichnungen, berechnete oder festgelegte Werte oder zur Darstellung benötigte leere Einträge.
 - `__getModulTotale`
Ruft basierend auf der erhaltenen PersonenId die noch benötigten Daten für die Studententotale des Dozenten aus der Datenbank ab und ergänzt sie mit den Daten aus `__getStundenTotale`, sowie festgelegten oder berechneten Werten oder zur Darstellung benötigte leere Einträgen. Die Daten werden anschliessend in einem Array gespeichert.
 - `__getSaldoData`
Ruft basierend auf der erhaltenen PersonenId die noch benötigten Daten für das Saldo des Dozenten aus der Datenbank ab und ergänzt sie mit den Daten aus `__getStundenTotale`, sowie festgelegten oder berechneten Werten oder zur Darstellung benötigte leere Einträgen. Die Daten werden anschliessend in einem Array gespeichert.
- **ShowingData**
Diese Klasse ist für das darstellen der Daten zuständig. Sie erstellt die Tabellen, formatiert sie korrekt und Speichert die pdf-Datei ab.
Dies bewerkstelligt sie mittels folgender Funktionen:
 - `__setFormat`
Legt alle benötigten Formatierungsdaten fest und speichert sie.
 - `__getFileName`
Diese Funktion ruft auf Grund der PersonenId den Personenkürzel der betreffenden Person ab. Falls dieser nicht vorhanden ist, ruft sie den Nach- und Vornamen der Person ab und speichert diese Informationen als FileName.
 - `__showData`
Erstellt die benötigten Tabellen auf Grund der von `GetDozData` erhaltenen Daten und den Formatierungsdaten welche in der Funktion `__setFormat` erstellt wurden. Die Funktion enthält ausserdem die Funktion:
 - `__drawPage`
Diese Funktion zeichnet auf jeder Seite die Fusszeile.

2.5.3 Chronologischer Ablauf

Hier wird der grobe chronologische Ablauf des Tools dargestellt:

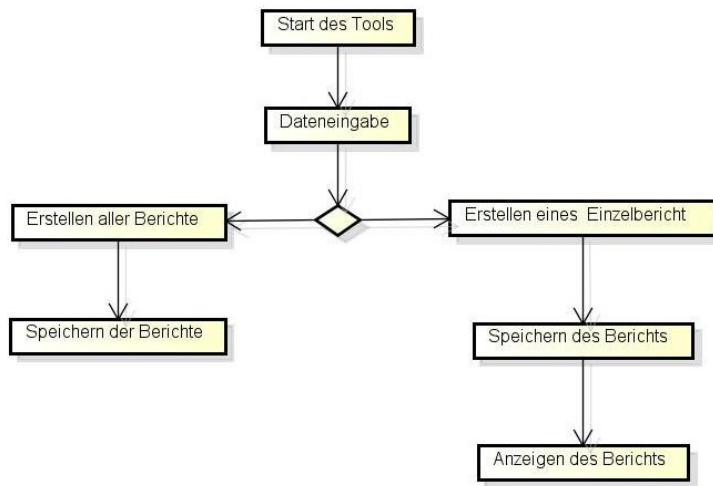


Abbildung 12 Chronologischer Ablauf

3 Realisierung

In diesem Kapitel werden einige Schlüsselstellen des Codes aufgezeigt. Genauere Informationen können dem Code selber oder dem dazugehörigen PyDoc entnommen werden.

3.1 Aufruf der Generierung

Hier geht es darum, auf Grund der Eingabe im GUI die entsprechenden Funktionen aufzurufen:

```
self.bClose.clicked.connect(self.__close)
self.bAlleBerichte.clicked.connect(self.__alleBerichte)
self.bEinzelBericht.clicked.connect(self.__einzelBericht)
```

Diese Funktionen konstruieren anschliessend die benötigten Klassen und übergeben diesen die benötigten Werte aus dem GUI.

3.2 Abruf der benötigten Personen

Die PersonenIds der benötigten Personen werden mittels zweier Abfragen abgerufen:

- Alle Personen, welche zur Kostenstelle „T IMES“ gehören:
`select PersId from person where kostenstelle_KostenstName = 'T IMES';`
- Alle Personen, welche in einem Modul mit der Kostenstelle „T IMES“ unterrichten.
`select s.person_PersId from anlass a, stunden_modul s, person p, modul m
where s.person_PersId = p.PersId and s.anlass_AnlassId = a.AnlassId and
a.modul_ModulKuerzel = m.ModulKuerzel and m.kostenstelle_KostenstName = 'T
IMES'`

Anschliessend werden die Ergebnisse dieser Abfragen zusammengelegt. Da es hierbei zu Duplikaten kommen kann, werden diese herausgefiltert:

```
for m in rowsPersonen_Dup:
    if m not in rowsPersonen:
        rowsPersonen.append(m)
```

3.3 Abrufen der Personen-, Stunden und Saldo-Daten

Hier wird der Array mit den PersonenIds Eintrag für Eintrag durchgegangen und es wird überprüft, welche dieser Dozenten überhaupt Stunden geleistet haben:

Dazu wird erst das Studenttotal des betreffenden Dozenten abgerufen:

```
select SUM(stundenModAnzahl) from stunden_modul where person_PersId = [PersonenId]
```

Um anschliessend zu überprüfen ob diese Abfrage ein Ergebnis lieferte:

```
if not dataStundenTotal == [None]:
```

Sollte ein Ergebnis erhalten worden sein, werden nun die Funktionen zum Abruf und Erstellen der benötigten Daten aufgerufen und die Daten in einer zur Datendarstellung idealen Art gespeichert. Hier ein Beispiel wie der mehrdimensionale Array mit den Personendaten aussieht:

```
[['Leistungsplanung: 13/14 - 01. August 2013 - 31. Juli 2014', None, None, None, None, None, None, None, None], [None, None, None, None, None, None, None, None, None], ['Name/Vorname', u'██████████', u'██████████', 'Personal-Nr.', '██████████', None, None, 'BG-V(%)', 100.0], ['Geburtsdatum', '██████████', None, 'Kurzzeichen', u'hsbh', None, None, 'BG-L(%)', 100.0], ['Kostenstelle', '██████████', u'T IMES', 'Ferienpauschale in Stunden (bez. BG-L)', None, None, None, None, 210.0], ['Eintritt', '██████████', None, 'Weiterbildungspauschale in Stunden (bez. BG-L)', None, None, None, None, 168.0], [None, None, None, None, None, None, None, None, None], ['Bruttoarbeitszeit', 2088.66, 'bei BG(100%)', 'Bruttoarbeitszeit (bez. BG-L)', None, None, 'Total Stunden', None, 2088.66], ['Saldo-Vorjahr', 295.04, None, 'Nettoarbeitszeit (bez. BG-L)', None, None, 'Total Stunden', None, 1906.8], [None, None, None, None, None, None, None, None, None]]
```

3.4 Darstellung der Daten

3.4.1 Tabellen

Bei der Darstellung der Daten werden verschiedene Tabellen erstellt. Die Formatierung dieser Tabellen werden in verschiedenen Arrays festgehalten, hier ein Beispiel:

```
self.formatTotalData = ([('BACKGROUND', (1,0),(4,0),colors.grey),  
                          ('BACKGROUND', (0,2),(5,3),colors.lightyellow),  
                          ('BACKGROUND', (5,0),(5,0),colors.lavenderblush),  
                          ('GRID', (0,0),(5,0),0.5,colors.black),  
                          ('BOX', (0,2),(5,3),0.5,colors.black),  
                          ('FONTNAME', (0,0),(0,0), 'Helvetica-Bold'),  
                          ('ALIGN', (1,0),(5,0), 'CENTER')])
```

Diese Formatierungs-Arrays können den Tabellen bei der Erstellung übergeben werden.

Die verschiedenen Tabellen werden anschliessend einem Array hinzugefügt, welcher bei der Erstellung des Pdfs aufgerufen wird.

3.4.2 Fusszeile

Die Fusszeile wird auf Grund bestimmter Koordinaten auf jede Seite gezeichnet:

```
self.date = time.strftime("%d.%m.%Y")  
canvas.saveState()  
canvas.setFillColor(gray)  
canvas.setFont('Helvetica',9)  
canvas.drawString(20, 15, "" + (self.fileName) + ".pdf")  
canvas.drawString(500, 15, "Datum: %s" %(self.date))  
canvas.restoreState()
```

3.4.3 Dateiname

Die Daten für den Dateinamen werden aus der Datenbank abgerufen. Da nicht bei allen Personen alle Daten vorhanden sind gibt es zwei Optionen für den Dateinamen, wobei die zweite nur aufgerufen wird, wenn die erste kein Ergebnis lieferte:

- Den Personenkürzel
select PersKuerzel from person where PersId = [PersonenId]
- Den Personen Vor- und Nachnamen
select PersNachname, PersVorname from person where PersId = ?

3.5 Geleisteter Code

Der für dieses Projekt erstellte Code kann separat aus dem Anhang entnommen werden.

3.6 Code-Kommentierung

Der Code wurde kommentiert.

4 Testing

Hier wird festgehalten, welche Testfälle getestet wurden, und wie deren Ergebnis aussah. Es gilt jedoch festzuhalten, dass während dem Realisierungsprozess immer wieder kleine Tests durchgeführt wurden.

4.1 Testmethode

Das Testen wurde mit den Dozentenberichten zweier Dozenten durchgeführt:

- hsbh
Bei diesem Dozenten sind nahezu alle Daten vorhanden.
- Ebei
Bei diesem Dozenten sind sehr wenig Daten vorhanden

Die ursprünglichen und die von diesem Tool ausgegebenen Dozentenberichte finden Sie im Ordner „IPA_JonBrunke\Testing“, wobei die ursprünglichen mit „_Old“ und die von diesem Tool mit „_New“ gekennzeichnet sind.

4.2 Testen während der Realisierung

Während dem Realisieren des Projekts wurden laufend Tests ausgeführt. Dazu gehörten etwa Testfälle wie:

- Werden die richtigen Daten ausgelesen?
- Werden berechnete, festgelegte, etc. Werte den vorhandenen Daten korrekt hinzugefügt?
- Werden Daten korrekt gespeichert?
- Werden die Daten korrekt auf dem Pdf abgebildet?
- Ist die Darstellung der Daten auf dem Pdf den Vorstellungen entsprechend?
- Etc.

Dadurch war das Ergebnis der meisten Testfälle schon während der Realisierung bekannt. Diese wurden jedoch trotzdem in das Testprotokoll aufgenommen.

4.3 Testfälle

In der folgenden Tabelle sind die zu testenden Testfälle abgebildet:

Testfall		Erwartetes Ergebnis
Personendaten		
Daten		
Die Daten-Bezeichnungen sind korrekt		Die Daten sind korrekt
Die aus der Datenbank erhaltenen Daten sind korrekt		Die Daten sind korrekt
Die festgelegten Daten sind korrekt		Die Daten sind korrekt
Die berechneten Daten sind korrekt		Die Daten sind korrekt
Darstellung		
Die Schrift-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Farb-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Positions-formatierung ist korrekt		Die Formatierung ist korrekt
Moduldaten		
Daten		
Die Daten-Bezeichnungen sind korrekt		Die Daten sind korrekt
Die aus der Datenbank erhaltenen Daten sind korrekt		Die Daten sind korrekt
Die festgelegten Daten sind korrekt		Die Daten sind korrekt
Die berechneten Daten sind korrekt		Die Daten sind korrekt
Darstellung		
Die Schrift-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Farb-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Positions-formatierung ist korrekt		Die Formatierung ist korrekt
Totaldaten		
Daten		
Die Daten-Bezeichnungen sind korrekt		Die Daten sind korrekt
Die aus der Datenbank erhaltenen Daten sind korrekt		Die Daten sind korrekt
Die festgelegten Daten sind korrekt		Die Daten sind korrekt
Die berechneten Daten sind korrekt		Die Daten sind korrekt
Darstellung		
Die Schrift-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Farb-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Positions-formatierung ist korrekt		Die Formatierung ist korrekt
Saldodaten		
Daten		
Die Daten-Bezeichnungen sind korrekt		Die Daten sind korrekt
Die aus der Datenbank erhaltenen Daten sind korrekt		Die Daten sind korrekt
Die festgelegten Daten sind korrekt		Die Daten sind korrekt
Die berechneten Daten sind korrekt		Die Daten sind korrekt
Darstellung		
Die Schrift-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Farb-formatierung der Daten ist korrekt		Die Formatierung ist korrekt
Die Positions-formatierung ist korrekt		Die Formatierung ist korrekt
Sonstiges		
Die Fusszeilen-Daten sind korrekt		Die Daten sind korrekt
Die Positions-formatierung der Fusszeile ist korrekt		Die Formatierung ist korrekt
Die Fusszeilen-formatierung ist korrekt		Die Formatierung ist korrekt

Tabelle 4 Testing - Testfälle

4.4 Testbericht

Diese Tabelle zeigt die erlangten Resultate beim Testen:

Testfall	Erwartetes Ergebnis	Erhaltenes Ergebnis	Erfüllt
Personendaten			
Daten			
Die Daten-Bezeichnungen sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die aus der Datenbank erhaltenen Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die festgelegten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die berechneten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Darstellung			
Die Schrift-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Farb-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Positions-formatierung ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Moduldaten			
Daten			
Die Daten-Bezeichnungen sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die aus der Datenbank erhaltenen Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die festgelegten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die berechneten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Darstellung			
Die Schrift-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Farb-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Positions-formatierung ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Totaldaten			
Daten			
Die Daten-Bezeichnungen sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die aus der Datenbank erhaltenen Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die festgelegten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die berechneten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Darstellung			
Die Schrift-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Farb-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Positions-formatierung ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Saldodaten			
Daten			
Die Daten-Bezeichnungen sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die aus der Datenbank erhaltenen Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die festgelegten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die berechneten Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Darstellung			
Die Schrift-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Farb-formatierung der Daten ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Positions-formatierung ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Sonstiges			
Die Fusszeilen-Daten sind korrekt	Die Daten sind korrekt	Die Daten sind korrekt	
Die Positions-formatierung der Fusszeile ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	
Die Fusszeilen-formatierung ist korrekt	Die Formatierung ist korrekt	Die Formatierung ist korrekt	

Tabelle 5 Testing - Testbericht

4.5 Testauswertung

Dadurch, dass bereits während der Realisierung viel getestet wurde, konnten alle Fehler bereits behoben werden. Somit ist nun keine weitere Korrektur nötig.

Anhang

Die Anhänge befinden sich auf der mit gelieferten CD. Dabei diese beinhaltet folgende Verzeichnisse, Unterverzeichnisse und Dateien:

- Dokumentation
- Testing
 - ebei_New.pdf
 - ebei_Old.pdf
 - hsbh_New.pdf
 - hsbh_Old.pdf
- Code
 - Gesamter Code
 - Dozententool.py
 - Geleisteter Code
 - GeleisteterCode.py
- Tool
 - Dozententool.py
- Testdaten
 - Planungstool_pentaho.xlsx