

Gesellschaft für Personalentwicklung und Bildung mbH

Projektdokumentation

Auftragsbearbeitungs Software für die Firma SECUDRIVE GmbH

Michael Lappenbusch

13.10.2014

Inhalt

1. Situationsanalyse	S.3
2. Aufgabenbeschreibung	S.3
3. IST Analyse	S.3
4. SOLL Konzept	S.4
5. Projektziele	S.5
6. Projektstrukturplan (PSP)	S.6
7. Zeitplan	S.7
8. Kostenplan	S.7
9. Qualitätssicherungsplan	S.8
10. Durchführung	S.9
11. Abschlussphase	S.12
12. Fachlicher Soll Ist Vergleich	S.12
13. Zeitlicher Soll Ist Vergleich	S.13
14. Fazit	S.13

SITUATIONSANALYSE

Nachdem ich der Firma SECUDRIVE GmbH die Möglichkeit der Auftragsbearbeitungs-Software vorgestellt habe sind Herr Linke und Herr Reich überzeugt dass dieses Produkt den Anforderungen gerecht wird.

Da die Firma SECUDRIVE GmbH die Produktivität auch in anderen kaufmännischen Bereichen steigern möchte, ist Herr Lehmann dafür dass die Arbeiten im Bereich der Auftragsbearbeitung automatisiert werden. Deshalb wurde die Entwicklung der Auftragsbearbeitungs-Software bei mir in Auftrag gegeben.

Ich werde nun die Anforderungen spezifizieren, mit Hilfe von UML-Diagrammen modellieren, das Projekt planen und es in C# umsetzen.

AUFGABENBESCHREIBUNG

Es soll eine erweiterbare Softwarelösung für die Auftragsbearbeitung entwickelt werden.

Die Auftragsbearbeitung soll die Bereiche Angebotserstellung, Auftragsfertigung und Rechnungslegung automatisieren, um den bisherigen Papieraufwand zu reduzieren.

Bei der Entwicklung der Software müssen folgende Richtlinien und Standards berücksichtigt werden:

Richtlinie über den Inhalt von Aufträgen, Richtlinie zur Rechnungserstellung, Richtlinie über den Aufbau von Bildschirmarbeitsplätzen.

Es soll eine reibungslose Bedienung der Kunden genauso möglich sein, wie die telefonische Annahme von Aufträgen.

IST ANALYSE

Aufträge werden schriftlich oder mündlich per Hand erfasst.

Es gibt bisher 5 Listen für sämtliche Daten:

Artikelliste, Angebotsliste, Auftragsliste, Kundenliste und Rechnungsliste.

Die Artikelliste enthält alle vorhandenen Artikel.

Wenn ein Kunde an einem Artikel interessiert ist wird für ihn ein Angebot erstellt, dass angenommen oder abgesagt werden kann.

Der Kunde wird in die Kundenliste eingetragen wenn er ein Angebot annimmt und noch nicht als Kunde eingetragen ist.

Danach wird der Auftrag in die Auftragsliste eingetragen.

Der Auftrag wird ohne besondere Formblätter oder Vordrucke entgegengenommen und in den Rechner eingegeben.

Darauf folgt eine mündliche oder schriftliche Auftragsbestätigung, bei der sich an einen Standardtext für die Auftragsabwicklung gehalten wird.

Nach der Erledigung des Auftrags bekommt der Kunde seine Rechnung.

Die Rechnung enthält: die Kundennummer, die Auftragsnummer, die Kundenanschrift, die

Lieferanschrift, die Rechnungsanschrift, die Position, den Gesamtpreis und die Zahlungsfrist.
Außerdem sind folgende Richtlinien vorhanden:
Richtlinie über den Inhalt von Aufträgen, Richtlinie zur Rechnungserstellung, Richtlinie über den Aufbau von Bildschirmarbeitsplätzen.

SOLL KONZEPT

Die C#-Anwendung soll effizient, zuverlässig, änderbar und wartbar sein.
Die Anpassungsfähigkeit bei Hardware-Austausch soll gegeben sein.
Der Ablauf soll so organisiert sein, dass der bisherige Ablauf der Kundenbedienung und der Stammdatenpflege kundenfreundlich durchgeführt werden kann.
Die bisherigen Wartezeiten sollen vermindert werden.
Vor der Projektanbahnung soll die Funktion des Programmsystems direkt in der Auftragsabwicklung getestet worden sein.
Die unter "Kurzform der Aufgabenstellung" genannten Richtlinien und Standards sollen eingehalten worden sein.

Die zu entwickelnde Software soll folgende Anforderungen erfüllen:

Bedienung der Kundendaten:

- Auftragsannahme
- Auftragsstornierung
- Angebotserstellung
- Rechnungserstellung

Stammdatenpflege:

- Pflege der Kundenliste
 - Kunde hinzufügen
 - Kunde bearbeiten
 - Kunde löschen
- Erstellung der Preisliste
- Artikel hinzufügen
- Artikel löschen

Die Artikeltable der Datenbank soll folgende Daten beinhalten:

Anzahl, Artikelnummer, Artikelbezeichnung und Preis.

Die Angebotstabelle der Datenbank soll folgende Daten beinhalten:

Angebotsnummer, voraussichtliches Lieferdatum, Bindungsfrist, Artikel, Anzahl und Preis.

Die Auftragsstabelle der Datenbank soll folgende Daten beinhalten:

Auftragsnummer, voraussichtlicher Liefertermin, Artikel, Anzahl, Kundennummer und Lieferanschrift.

Die Kundentabelle der Datenbank soll folgende Daten beinhalten:

Kundennummer, Kundenname, Kundenanschrift, Rechnungsanschrift, Lieferanschrift und weitere Eigenschaften.

Die Rechnungstabelle der Datenbank soll folgende Daten beinhalten:
Auftragsnummer, Kundennummer, Gesamtpreis und Zahlungsfrist.

In den frühen Entwicklungsphasen soll ein Prototyp generiert werden.

Im Benutzerhandbuch sollen folgende Sicherheitsanforderungen genau erklärt werden:
Sicherheitskopien von den geänderten und neuen Daten sind am Ende jedes Arbeitstages anzufertigen.

Bei Ausfall der Rechentechnik ist die Auftragsabwicklung manuell weiterzuführen, und nach der Wiederherstellung der Funktionstüchtigkeit sind die Daten zu übertragen.

Die Anwendung soll in der Programmiersprache C# erstellt werden, hierbei soll die Entwicklungsumgebung Visual Studio 2010 genutzt werden.
Die zu entwickelnde Software soll unter Windows 7 laufen.

PROJEKTZIELE

SACHZIEL

Es soll eine effiziente, änderbare und wartbare Auftragsbearbeitungs-Software entwickelt werden.

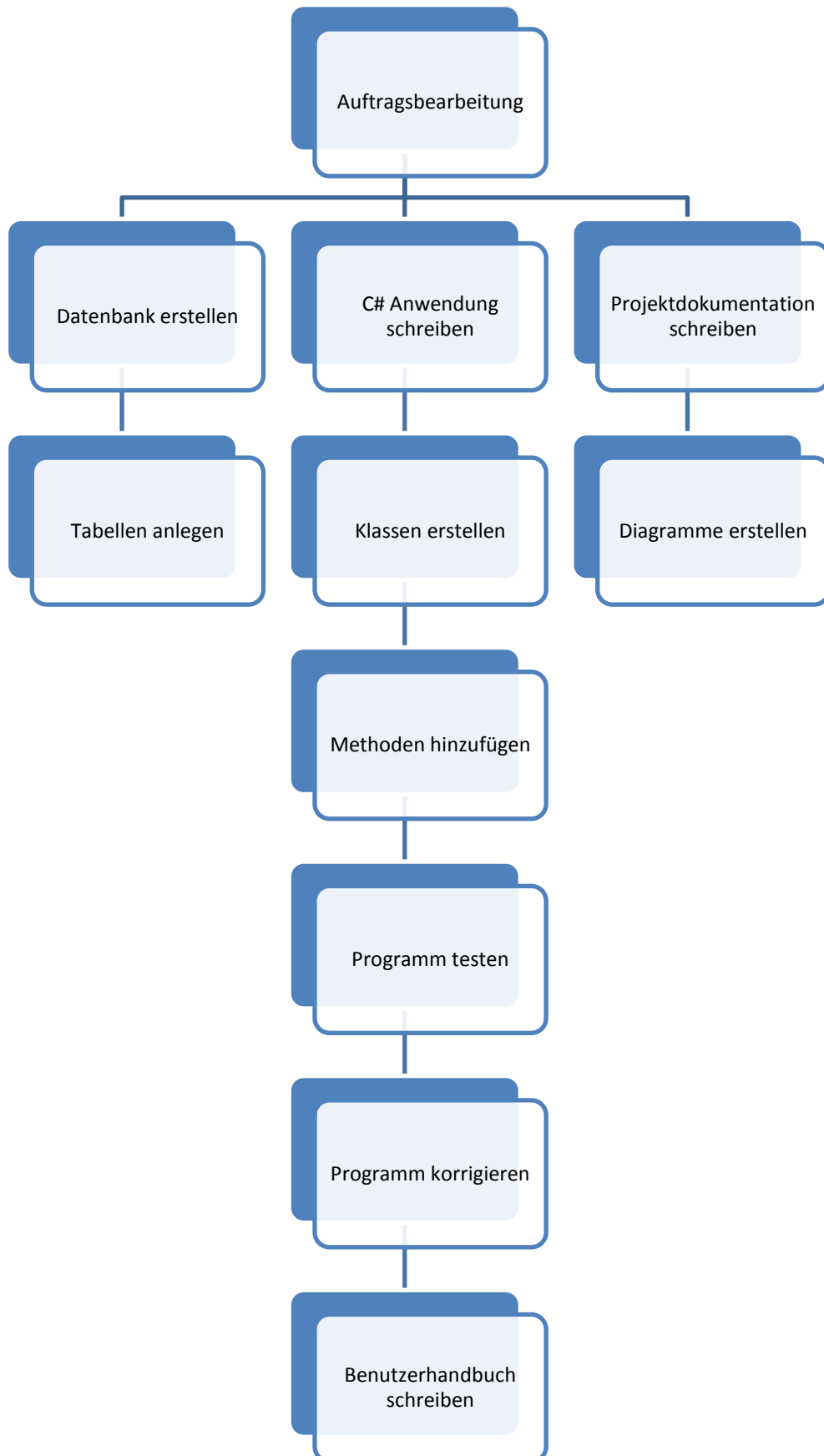
TERMINZIEL

Das Projekt soll bis zum 24.10.2014 fertiggestellt sein.

KOSTENZIEL

Die Kosten sollen so gering wie möglich gehalten werden.

PROJEKT STRUKTUR PLAN (PSP)



ZEITPLAN

Prozess	Teilprozess	Stunden
Definitionsphase	Projektwahl	2
10 Stunden	Ideenfindung	3
	Projektanforderungen festlegen	5
Planungsphase	PSP erstellen	1
20 Stunden	Terminplan	3
	Kostenplan	3
	Qualitätsplan	3
	UML - Diagramm	3
	Internetrecherche	7
Implementierungsphase	Projektskizze	2
	Benutzeroberfläche	
30 Stunden	Prototyp programmieren	15
	Prototyp testen	4
	Prototyp Fehler beheben	8
	Klassendiagramm erstellen	1
Abschlussphase	Benutzerhandbuch schreiben	1
10 Stunden	Dokumentation	8
	Abgabe	1
	GESAMTDAUER	70

KOSTENPLAN

Personalkosten			
Personal	Stunden	Stundenlohn	Gesamtlohn
Praktikant	70	0,00€	0,00€
Summe			0,00€
Fixkosten Pauschal			
Kostenart	Tagespreis	Stundenanzahl	Gesamtkosten
Bürokosten	14,50€	70	126,86€
Kosten Arbeitsplatz PC	0,65€	70	5,68€
Summe			132,54€
Variable Kosten Pauschal			
Kostenart	Tagespreis	Stundenanzahl	Gesamtkosten
Materialkosten	0,42€	70	3,66€
Summe			3,66€
Projektgesamtkosten			136,20€

QUALITÄTSSICHERUNGSPLAN

Es wird zuerst ein ERD (Entity Relationship Modell) angefertigt, durch welches ersichtlich ist welche Tabellen (Zeilen und Spalten) in der Datenbank benötigt werden und in welchem Verhältnis diese zu einander stehen.

Als nächstes wird ein UML-Diagramm angefertigt auf dem klar zu sehen ist welche Arbeitsabläufe der Mitarbeiter (Telefonist) beim der Auftragsbearbeitung hat. Auf Grund dieses UML-Diagramms wird festgelegt welche Funktionen für das Programm benötigt werden.

Danach wird die Benutzeroberfläche mit Visual Studio in C# erstellt, in der 1. Form wird für jeden erforderlichen Arbeitsschritt ein Buttons erstellt, der jeweils eine eigene Form öffnet um den jeweiligen Arbeitsschritt umzusetzen.

Es wird eine Form geben die alle benötigten Tabellen in der Datenbank anlegt oder löscht, nachdem ein Administrations-Passwort eingegeben wurde.

Eine Form wird für die Erfassung von Neukunden sein, eine weitere zum bearbeiten der Kundendaten.

Desweiteren wird es eine Form zum Erfassen von Artikeln geben, sowie eine Form zum Bearbeiten der Artikel.

Darauf wird eine Form erstellt die zum erfassen der Aufträge dient, diese Form soll eine Funktion beinhalten welche die zu bestellenden Artikel mit den gelagerten Artikeln abgleicht und ggf. eine Fehlermeldung ausgibt wenn die Bestellmenge nicht mehr verfügbar ist.

Sobald das Angebot gespeichert ist soll die Stückzahl der Artikel vom Lagerbestand abgezogen werden.

Danach wird eine Form erstellt die zum Stornieren der Aufträge dient, diese Form löscht erstellte Aufträge wieder aus der Datenbank und fügt die Stückzahlen des Auftrags den Artikelmenge wieder hinzu.

Als nächstes wird eine Form erstellt mit der Angebote versendet werden können, dort wird ein Angebot erfasst, gespeichert und als Textdatei ausgegeben, die folglich gedruckt oder per Email versendet werden kann. Der Kopf der Textdatei soll wie bei allen Textdateien die noch erzeugt werden sollen immer den Empfänger sowie der Absender enthalten.

Angebotsannahme ist die Aufgabe der nächsten Form, in ihr wird das Angebot ausgewählt für das sich ein Kunde entschieden hat. Dieses Angebot wird darauf verbucht und als Rechnung mit Kontodaten ausgegeben, wieder als Textdatei die gedruckt und/oder versendet werden kann. Die Daten der Tabellenspalte des Angebots sollen aus der Datenbank gelöscht werden nachdem sie der Tabelle Rechnungen hinzugefügt wurden.

Die nächste Form nennt sich Preisliste versenden, in ihr soll man den Bestandskunden auswählen können, nach einem Buttonklick soll eine Textdatei mit der Preisliste aller Artikel ausgegeben werden.

Die letzte Form soll für das Erstellen der Rechnungen sein, in ihr wird ein Auftrag ausgewählt und als Rechnung mit Kontodaten wie zuvor als Textdatei ausgegeben. Darauf werden die Rechnungsdaten in die Datenbank eingetragen und die Daten des Auftrags gelöscht.

Zuletzt wird der InfoButton mit einer MessageBox belegt, die meine Kontaktdaten als Ansprechpartner für dieses Programm ausgibt.

DURCHFÜHRUNG

Ich habe ein ERD erstellt in das ich alle Tabellen die verwendet werden sollen eingetragen habe. Zudem habe ich ein UML-Diagramm angefertigt in dem ersichtlich ist welche Aufgaben der Mitarbeiter (Telefonist) in Zukunft mit der von mir geschriebenen Software bewerkstelligen muss. Ich habe in Visual Studio ein neues WindowsForms Projekt erstellt und es Auftragsbearbeitung genannt. Als erstes habe ich die 1.Form erstellt, das Hauptfenster von dem aus das Programm organisiert wird. Darauf habe ich 12 Buttons dieser 1.Form hinzugefügt und so beschriftet das man sofort erkennt welche Arbeitsschritte sich dahinter verbergen sollen.

Als nächstes habe ich eine Form für das Administrieren der Datenbank erstellt, sie öffnet sich wie alle Formen der Anwendung in der Mitte des Bildschirms.

Diese Form beinhaltet eine Textbox in welche ein von mir vorgegebenes (und editierbares) Passwort eingegeben werden soll. Das Passwort ist bei der Eingabe nicht sichtbar, und liegt bei Abgabe der Geschäftsleitung vor.

Unter der Textbox habe ich 2 Radiobuttons platziert, von denen nur einer ausgewählt werden kann. Der 1. Radiobutton sorgt für das löschen der Datenbank, der 2. Radiobutton hingegen für das erstellen der Datenbank, das Anlegen der Tabellen und das setzen sämtlicher Schlüssel mit denen die Tabellen unter anderem miteinander verbunden werden.

Unter den Radiobuttons habe ich einen Button platziert mit der Aufschrift Senden.

Nach einem Klick auf diesen Button wird das Passwort kontrolliert und danach die entsprechende Aktion ausgeführt. Danach erscheint eine MessageBox als Bestätigung, die Form schließt sich anschließend selbst.

Als nächstes habe ich die Form zur Neukunden Anmeldung angelegt und bestimmt das sie mit einem Klick auf den Button „Neukunden aufnehmen“ geöffnet wird.

Diese Form enthält Eingabefelder für sämtliche Kundendaten und Eingabefelder für einen Ansprechpartner, darunter habe ich einen Button mit der Beschriftung „Neukunde hinzufügen“ platziert. Ein Klick auf diesen Button bewirkt dass die eingegebenen Kundendaten in der Tabelle t_kunden der Datenbank gespeichert werden. Nach dem speichern der Daten erscheint eine MessageBox als Bestätigung, die Form schließt sich anschließend selbst.

Danach habe ich die Form Kundendaten bearbeiten angelegt und den Button mit gleicher Beschriftung entsprechend eingestellt, sodass sie sich öffnet wenn man drauf klickt.

Auf der linken Seite dieser Form befindet sich eine Listbox in der alle eingetragenen Kunden aufgelistet werden, ich habe diese Listbox so konfiguriert das sich sämtliche Kundendaten in den Eingabefeldern auf der rechten Seite zeigen, wenn man dort auf einen Kundennamen klickt.

Die Eingabefelder bleiben verschlossen (nicht editierbar) bis auf den Button mit der Aufschrift „Bearbeiten“ geklickt wird. Sobald der Button geklickt wurde öffnen sich die Eingabefelder und man kann die Daten bearbeiten. Nach der Bearbeitung klickt man dann auf der daneben liegenden Button „Speichern“ und die Kundendaten werden in der Datenbank verändert. Diese Form schließt sich anschließend selbst.

Die Form „Auftrag erfassen“ wurde dann als nächstes von mir angelegt und bei dem entsprechenden Button der Hauptform eingetragen.

Sie enthält Eingabefelder für den Artikelnamen, die Stückzahl und den Preis des Artikels.

Es wurde ein Label mit einem Hinweis darauf das der Preis mit einem Punkt anstelle eines Kommas eingegeben werden muss von mir darunter angebracht, damit es nicht zu einer Fehlermeldung kommt.

Sobald man dort alle Eingaben gemacht hat soll man auf den Button „Artikel hinzufügen“ klicken, den

ich darunter platziert habe. Der Artikel wird in der Datenbank gespeichert sobald dieser Button angeklickt wurde, es erscheint eine MessageBox als Bestätigung bevor sich die Form selbstständig schließt.

Anschließend habe ich die Form „Artikel bearbeiten“ erstellt und entsprechend eingebunden.

Diese Form enthält auf der linken Seite eine Listbox in der alle Artikel aufgelistet werden.

Die Listbox ist so konfiguriert das die Artikeldaten in den verschlossenen Eingabefeldern auf der rechten Seite erscheinen sobald ein Artikelname ausgewählt wurde.

Darunter befindet sich der Button „Bearbeiten“ der die Eingabefelder zum bearbeiten öffnet.

Nach dem bearbeiten soll man auf den Button „Speichern“ klicken der sich daneben befindet.

Sobald dieser Button geklickt wurde werden sämtliche Eingabedaten in der Datenbank geändert.

Diese Form eignet sich zur Bestandskontrolle des Lagers bzw. zur Lagerverwaltung.

Danach habe ich die passende Form für den Button „Auftrag hinzufügen“ erstellt und den Button so konfiguriert das sie sich mittig ohne Schließenbutton (dem X) öffnet.

Auf der linken Seite befindet sich eine Listbox in der sämtliche Kunden aufgelistet sind.

Direkt daneben eine Listbox in der alle Artikel aufgelistet sind, sowie eine Textbox für die eingabe der gewünschten Artikelstückzahl. Darunter befindet sich ein Button mit der Beschriftung „Position

hinzufügen“. Unter diesen Button habe ich eine Listbox platziert in der alle Positionen anhand ihrer Positionsnummer aufgelistet werden. Darunter befindet sich ein Button mit der Aufschrift „Fertig“.

Nachdem man in der linken Listbox den Kunden ausgewählt hat, in der Listbox daneben den Artikel

gewählt und in der Textbox die Stückzahl eingetragen hat. Klickt man den Button „Position hinzufügen“. Diesen Button habe ich so konfiguriert das zuerst die gewünschte Artikelstückzahl mit

der vorhandenen Stückzahl abgeglichen wird. Sollten zu wenig Artikel auf Lager sein öffnet sich eine MessageBox mit einer entsprechenden Meldung. Wenn genug Artikel auf Lager sind wird diese

Position in der Datenbank gespeichert, sowie der Auftrag in der Auftragsstabelle der Datenbank

erstellt. Die Artikelstückzahl wird in der Artikeltabelle um die entsprechende Anzahl verringert.

Darauf erscheint eine MessageBox mit einer Bestätigung und die Listbox mit den Kundennamen

verschwindet, da der Kunde des Auftrags nun feststeht. Es können als nächstes weitere Positionen

hinzugefügt werden. Mit einem Klick auf Fertig beendet man die Eingabe des Auftrags und das

Fenster schließt sich.

Darauf habe ich die Form „Auftrag stornieren“ angelegt und den Button der Hauptform so

konfiguriert das sie sich öffnet wenn er angeklickt wird.

Die Form enthält auf der linken Seite eine Listbox in der sämtliche Aufträge mit ihrer

Auftragsnummer vermerkt sind.

Sobald einer dieser Einträge angeklickt wird erscheinen in den Textboxen auf der rechten Seite

Kundennummer, Kundenname und das Auftragsdatum. Daneben befindet sich ein Button mit der

Aufschrift „STORNIEREN“. Nachdem dieser Button geklickt wird werden in der Datenbank der

entsprechende Auftrag und seine Positionen gelöscht. Die Artikelstückzahlen werden in der

Artikelliste beim entsprechenden Artikel wieder hinzugefügt. Darauf schließt sich diese Form selbst.

Im Anschluss darauf habe ich die Form „Angebot versenden“ erstellt und entsprechend eingebunden.

Sobald sich diese Form öffnet sieht man links eine Listbox in der man einen Kunden auswählen soll,

rechts daneben habe ich eine weitere Listbox mit sämtlichen Artikeln platziert. Daneben befindet

sich eine Textbox für die Artikelstückzahl und eine Textbox in welche der Artikelpreis für dieses

Angebot eingetragen werden soll. Unter dieser Textbox befindet sich ein Label als Hinweis darauf das man bei der Preiseingabe einen Punkt anstelle eines Kommas benutzen soll.

Darunter befindet sich ein Button mit der Aufschrift „Angebot erstellen“.

Sobald Kunde und Artikel gewählt sind, und Stückzahl sowie Preis eingegeben wurden, soll der

Button geklickt werden.

Sobald dieser Button ausgelöst wurde, wird die Artikelstückzahl mit der in der Datenbank gespeicherten Stückzahl verglichen. Im Falle dass zu wenig Artikel gelagert sind erscheint eine Fehlermeldung die auf die Beschaffung neuer Ware hinweist.

Wenn genug Artikel vorhanden sind, wird das Angebot in der Datenbank gespeichert und eine Textdatei im Verzeichnis „C:\Auftragsbearbeitung\Angebote\“ erzeugt diese Textdatei enthält die Kundennummer im Dateinamen.

Die Textdatei beinhaltet Empfänger und Absender einen standarttext für das Angebot, sowie das eigentliche Angebot. Sie kann einfach ausgedruckt oder per Email verschickt werden.

Anschließend habe ich die Form für den Button „Angebotsannahme“ erstellt und entsprechend eingebunden. Diese Form enthält eine Listbox mit sämtlichen Angebotsnummern, sobald eine dieser Nummern ausgewählt wurde erscheint zur Kontrolle der Kundennamen in der sich darunter befindenden Textbox. Als Warnhinweis habe ich darunter ein Label erstellt welches darauf hinweist das beim erstellen der Rechnung das Angebot gelöscht wird. Unter diese, Label habe ich einen Button mit der Beschriftung „Rechnung erstellen“ platziert. Sobald dieser Button angeklickt wird schließt sich die Form selbst, es werden die Angebotsdaten in die Datenbanktabelle t_rechnung geschrieben und es wird eine Rechnung in Form einer Textdatei ausgegeben. Diese Textdatei befindet sich im Verzeichnis „C:\Auftragsbearbeitung\Rechnungen\“ und hat die Kundennummer im Dateinamen.

Darauf habe ich eine Form für den Button „Preisliste versenden“ erstellt und korrekt eingebunden. Diese Form besteht aus einer Listbox in der alle Kunden aufgelistet werden, darunter befindet sich ein Button mit der Aufschrift „Preisliste erstellen“.

Sobald dieser Button angeklickt wurde schließt sich die Form selbst und es erscheint eine Textdatei die Empfänger und Absenderdaten enthält sowie eine Auflistung sämtlicher Artikel, Stückzahlen und Preise. Diese Textdatei befindet sich im Verzeichnis „C:\Auftragsbearbeitung\Preislisten\“ und hat die Kundennummer im Dateinamen.

Danach habe ich die Form „Rechnung versenden“ erstellt und passend eingebunden.

Diese Form enthält eine Listbox in der sämtliche Auftragsnummern aufgelistet werden, darunter befindet sich eine Textbox in welcher der Kundennamen zur Kontrolle nach dem auswählen erscheint. Unter diese Textbox habe ich ein Label als Warnhinweis platziert, dass besagt das der Auftrag nach dem erstellen der Rechnung gelöscht wird.

Darunter befindet sich ein Button mit der Aufschrift „Rechnung versenden“.

Sobald dieser Button angeklickt wird werden sämtliche Daten des Auftrags in der Tabelle t_rechnung der Datenbank gespeichert, worauf der Auftrag gelöscht wird. Darauf schließt sich die Form selbstständig und es erscheint die fertige Rechnung in einer Textdatei mit Kundennummer im Dateinamen die im Verzeichnis „C:\Auftragsverarbeitung\Rechnungen\“ gespeichert ist.

Als letztes habe ich den Button „Info“ der Hauptform mit einer MessageBox belegt, die nach einem Klick erscheint. In dieser MessageBox stehen meine Kontaktdaten als Ansprechpartner für diese Software. Ein Benutzerhandbuch wurde von mir verfasst und der Software beigelegt.

ABSCHLUSSPHASE

Im Abschlussgespräch habe ich die Software anhand einer Präsentation erklärt, wo ich auf sämtliche Arbeitsabläufe genau eingegangen bin.

Die Geschäftsleitung war sehr zufrieden mit meiner Arbeit und es wurde mir zugesichert das man auf mich zukommen wird wenn neue Projekte anstehen.

FACHLICHER SOLL IST VERGLEICH

Im Abschlussgespräch habe ich die Software anhand einer Präsentation erklärt, wo ich auf sämtliche Arbeitsabläufe genau eingegangen bin

Teilprozess	Ja	Nein
Projektwahl	✓	
Ideenfindung	✓	
Projektanforderungen festlegen	✓	
PSP erstellen	✓	
Terminplan	✓	
Kostenplan	✓	
Qualitätsplan	✓	
UML - Diagramm	✓	
Internetrecherche	✓	
Projektskizze Benutzeroberfläche	✓	
Prototyp programmieren	✓	
Prototyp testen	✓	
Prototyp Fehler beheben	✓	
Klassendiagramm erstellen	✓	
Benutzerhandbuch schreiben	✓	
Dokumentation	✓	
Abgabe	✓	

ZEITLICHER SOLL IST VERGLEICH

Prozess	Teilprozess	Soll	Ist
Definitionsphase	Projektwahl	2	1
10 Stunden	Ideenfindung	3	4
	Projektanforderungen festlegen	5	5
Planungsphase	PSP erstellen	1	1
20 Stunden	Terminplan	3	3
	Kostenplan	3	2
	Qualitätsplan	3	3
	UML - Diagramm	3	2
	Internetrecherche	7	9
Implementierungsphase	Projektskizze	2	2
	Benutzeroberfläche		
30 Stunden	Prototyp programmieren	15	18
	Prototyp testen	4	3
	Prototyp Fehler beheben	8	6
	Klassendiagramm erstellen	1	1
Abschlussphase	Benutzerhandbuch schreiben	1	1
10 Stunden	Dokumentation	8	8
	Abgabe	1	1
	GESAMTDAUER	70	70

FAZIT

Es ist zu einigen Problemen gekommen, da weder die Firmenadresse noch die Kontodaten der Firma SECUDRIVE GmbH im Projektleitfaden vermerkt war. Nach längerer Internetrecherche wurde ich dann allerdings im Internet fündig. Ein Stakeholder war zu der Zeit leider nicht erreichbar. Die Programmierung des Prototyps hat länger gedauert als erwartet, dafür sollte in Zukunft mehr Zeit einplanen.