



Abschlussprüfung Sommer 2025

Fachinformatiker Anwendungsentwicklung

Dokumentation der betrieblichen Projektarbeit

# **Implementierung von Buchungsstapel in der Financial Suite**

**Prüfungsbewerber:**

Lukas Schütz



Praktikumsbetrieb:

Versino AG CC Finance

Ulmer Landstraße 249

86391 Stadtbergen



## Implementierung von Buchungstapel in Financial Suite

---

### Inhaltsverzeichnis

Inhaltsverzeichnis .....	2
Abbildungsverzeichnis .....	3
Abkürzungsverzeichnis .....	3
1 Einleitung .....	4
1.1 Projektumfeld .....	4
1.2 Projektziel und Hintergründe .....	4
1.3 Projektschnittstellen .....	4
2 Projektplanung .....	5
2.1 Projektphasen .....	5
2.2 Ressourcenplanung .....	5
3 Analysephase .....	5
3.1 Ist-Analyse .....	5
3.2 Wirtschaftlichkeitsrechnung .....	5
4 Phase 1: Analyse und Anforderungsdefinition .....	7
4.1 Zielplattform .....	7
4.2 Anforderungsdokument .....	7
4.3 Visueller Aufbau des Forms .....	7
5 Phase 2: Entwicklung der Erweiterung .....	8
5.1 Implementierung der Benutzeroberfläche (Form) .....	8
5.2 Implementierung der Programmlogik .....	9
6 Phase 3 Testen und Anpassen .....	13
6.1 Funktionale und Logische Prüfung .....	13
7 Produktdokumentation .....	15
7.1 Anwenderdokumentation .....	15
7.2 Entwicklerinformation .....	15
8 Fazit .....	15
8.1 Soll-/Ist- Vergleich .....	15

## Implementierung von Buchungsstapel in Financial Suite

---

8.2	Gesammelte Erfahrungen .....	16
8.3	Ausblick für zukünftige Anpassungen .....	16
A	Anhang .....	17
A.1	Zeitliche Planung des Projekts .....	17
A.2	Verwendete Ressourcen .....	17
A.3	Anwenderdokumentation .....	18

## Abbildungsverzeichnis

Abbildung 1: Buchungsstapel-Layout .....	8
Abbildung 2: Soll-/Ist- Zeitplan Tabelle .....	16
Abbildung 3: Zeitplan-Tabelle .....	17

## Abkürzungsverzeichnis

ERP	Enterprise-Resource-Planning
SAP B1	SAP Business One
UI-API	User Interface - Application Programming Interface
DI-API	Data Interface - Application Programming Interface
SDK	Software Development Kit
CFL	Choose from List
UML	Unified Modeling Language
SQL	Structured Query Language
MSSQL	Microsoft SQL

## Implementierung von Buchungstapel in Financial Suite

---

# 1 Einleitung

## 1.1 Projektumfeld

Die frühere Versino Projects GmbH wurde im Jahr 2024 in die Versino AG integriert und bildet nun die Abteilung Finance. Die Versino AG positioniert sich seit über 22 Jahren als führender SAP Gold Partner für ERP-Softwarelösungen, die speziell auf die Bedürfnisse des Mittelstandes zugeschnitten sind. Ein engagiertes Team stellt als verlässlicher SAP Business One Partner die nahtlose Implementierung und den reibungslosen Betrieb von SAP B1 sowie angebundener Partnerlösungen sicher.

Im Gegensatz zu gängigen SAP Business One Partnern liegt der Schwerpunkt des Unternehmens auf der Entwicklung individueller Lösungen. Es zeichnet sich besonders durch die Realisierung umfangreicher Erweiterungen aus, sei es durch die Erstellung maßgeschneiderter Add-ons für spezifische Kundenanforderungen, durch Eigenentwicklungen wie die Financial Suite oder durch die Integration zusätzlicher Funktionen in bestehende Add-On-Lösungen, um individuellen Anforderungen bestmöglich gerecht zu werden.

Das aktuelle Projekt fokussiert sich auf die Erweiterung der Financial Suite, die zukünftig ein fester Bestandteil des Financial Suite Cockpits wird.

## 1.2 Projektziel und Hintergründe

Das Projektziel ist die Integration eines weiteren Forms in die Financial Suite. Dieses wird als neuer Funktionsbereich in das Financial Suite Cockpit eingebunden und soll die stapelweise Erfassung sowie anschließende, gesammelte Verbuchung von gleichen oder ähnlichen Geschäftsvorfällen ermöglichen. Der Hintergrund für dieses Vorhaben ist der hohe manuelle Aufwand des aktuellen Prozesses, bei dem für jede Buchung eine separate Transaktion manuell erstellt werden muss.

## 1.3 Projektschnittstellen

Bei der Entwicklung des Moduls wurden Funktionen des SAP Business One Software Development Kits (SDK) benutzt, sowie der eigens entwickelten VerXs Framework. Diese beiden Komponenten gewährleisten eine nahtlose Kommunikation zwischen der Financial Suite, der Anwendung und dem Server. Das VerXs Framework wurde von einem Mitarbeiter der ehemaligen Versino Projects GmbH entwickelt, um die Arbeit mit der „SAPbouiCOM“ und der „SAPbobsCOM“ zu erleichtern.

Die „SAPbouiCOM“ und „SAPbobsCOM“ sind beides Pakete der SAP Business One SDK; sie bilden die programmatische Schnittstelle zur UI-API (User Interface-Application Programming Interface) und der DI-API (Data Interface-Application Programming Interface). Das „SAPbouiCOM“ wird verwendet, um über die UI-API mit der Anwendung zu kommunizieren und ermöglicht bedingte Beeinflussungen und Anpassungen der Benutzeroberfläche. Es ermöglicht somit die Interaktion mit dem System, um bestimmte Anwendungsfunktionen zu steuern und anzupassen.

## Implementierung von Buchungstapel in Financial Suite

---

Hingegen dient die „SAPbobsCOM“ dazu, über die DI-API mit dem Server zu kommunizieren. Sie ermöglicht das programmatische Arbeiten mit SAP-Business-Objekten, um Geschäftsdaten zu erstellen, zu ändern oder abzurufen. Anstatt die Datenbank direkt zu manipulieren, stellt die DI-API sicher, dass jede Datenänderung (wie das Erstellen einer Buchung) die gesamte Geschäftslogik und die Berechtigungen des angemeldeten Benutzers durchläuft und protokolliert wird.

## 2 Projektplanung

### 2.1 Projektphasen

Das Projekt fand vom 08.05.2025 bis zum 21.05.2025 mit einer täglichen Arbeitszeit von 8 Stunden statt.

Folgende Projektphasen wurden innerhalb des Projektantrages definiert:

Siehe Anhang A.1 Zeitliche Planung des Projekts.

### 2.2 Ressourcenplanung

Im Anhang A.2 (Verwendete Ressourcen) sind sämtliche Ressourcen aufgelistet, die während des Projektes genutzt wurden, darunter Software, Hardware und Räumlichkeiten. Bei der Auswahl wurde konsequent darauf geachtet, auf bereits im Besitz der Versino AG befindliche oder kostenfreie verfügbare Mittel zurückzugreifen. Dieses Vorgehen gewährleistet eine effiziente und wirtschaftliche Projektdurchführung ohne zusätzliche finanzielle Belastungen.

## 3 Analysephase

### 3.1 Ist-Analyse

Zusammenfassend lässt sich feststellen, dass der Prozess sehr zeitaufwendig ist, da für jede Transaktion eine separate Buchung angelegt werden muss. Es fehlt eine Funktion, um mehrere Transaktionen gesammelt in einer Stapelbuchung zu verarbeiten.

Das angestrebte Ziel des Forms besteht darin, diesen zeitaufwendigen Prozess zu reduzieren, indem es die Möglichkeit bietet mehrere Transaktionen in form einer Stapelbuchung an das SAP-System zu übergeben.

### 3.2 Wirtschaftlichkeitsrechnung

Wie bereits in den Abschnitten 1.2 und 3.1 erwähnt und aufgrund aktueller Erfahrungswerte ist der Bedarf für eine solche Lösung festgestellt worden. In dem folgenden Abschnitt wird eine detaillierte Aufstellung der entstandenen Kosten und potenziellen Umsätze des Projekts näher erklärt.

## Implementierung von Buchungstapel in Financial Suite

---

Diese Analyse umfasst eine detaillierte Aufschlüsselung der finanziellen Aspekte, einschließlich der aufgewendeten Ressourcen, der Projektkosten sowie der erwarteten Umsätze aus dem Verkauf oder der Bereitstellung der entwickelten Lösung. Dabei werden die geschätzten Kosten für Entwicklung, Implementierung und Vermarktung der Lösung berücksichtigt.

Des Weiteren werden potenzielle Einnahmen oder Umsätze aus dem Verkauf bzw. Bereitstellung der Lösung an Kunden oder durch Dienstleistungen im Zusammenhang mit der Implementierung oder Anpassung erörtert.

Durch diese detaillierte Darstellung der entstehenden Kosten und potenziellen Einnahmen soll eine fundierte Einschätzung darüber ermöglicht werden, inwiefern sich das Projekt finanziell auswirkt und welchen Nutzen es für das Unternehmen in Bezug auf Umsatzgenerierung und Rentabilität hat.

Zur Bestimmung der Wirtschaftlichkeitsrechnung wurden hierbei folgende Variablen berücksichtigt:

**Ressourcennutzung/h** Betriebliche Ressourcen, sowie z.B. Räumlichkeiten, Lizenzkosten und Equipment werden mit einer stündlichen Pauschale von 35 € angerechnet.

**Personalkosten/h** Die betrieblichen Kosten für einen Umschüler werden hierbei mit einem Wert von 1 € eingerechnet. (Dieser Wert basiert auf dem monatlichen Entgelt von 200 €)

**Personalumsatz/h** Der durchschnittliche stündliche Umsatz eines Mitarbeiters beim normalen täglichen Geschäft läge hierbei bei 70 €.

**Produktpreis/Stück** Der Verkaufswert des aus dem Projekt entstandenen Moduls wird mit einem Wert von 250 € gerechnet.

**Marketingaufwand/Stück** Ein grob geschätzter Marketingkostenaufwand der voraussichtlich pro Kunde zu erbringen ist, läge hier bei etwa 100 €.

**Projektdauer** Der aktuell geplante zeitliche Aufwand für die Erstellung des Moduls beträgt 80 h, wovon 10 h für die Dokumentation vorgesehen sind.

Mit den zuvor genannten Daten ließen sich in Bezug auf das Projekt nun folgende Rechnung aufstellen:

1. **Ressourcennutzung/h + Personalkosten/h = Entwicklungskosten/h**
  - $35 \text{ €/h} + 1 \text{ €/h} = 36 \text{ €/h}$
2. **Produktpreis/Stück – Marketingaufwand/Stück = Produktwert/Stück**
  - $250 \text{ €/Stück} - 100 \text{ €/Stück} = 150 \text{ €/Stück}$
3. **Entwicklungskosten/h \* Projektdauer = Projektkosten**
  - $36 \text{ €/h} \times 70 \text{ h} = 2520 \text{ €}$
4. **Projektkosten / Produktwert/Stück = Mindest-Verkäufe für Kostengleich**
  - $2520 \text{ €} / 150 \text{ €/Stück} = 17 \text{ Stück (Aufgerundet)}$

Anhand der gezeigten Rechnung ist nun ersichtlich, dass für den Ausgleich der entstandenen Kosten der Verkauf des Moduls an mindestens 17 Kunden notwendig ist. Es ist davon auszugehen, dass das Ziel von 17 Kunden zu erreichen ist

## Implementierung von Buchungstapel in Financial Suite

---

### 4 Phase 1: Analyse und Anforderungsdefinition

#### 4.1 Zielplattform

Wie in Abschnitt 1.2 bereits erläutert, ist das Ziel des Projekts die Entwicklung eines neuen Forms für das Financial Suite Cockpit. In diesem Kontext ist ein solches Form eine eigenständige Benutzeroberfläche, die auf Basis der Windows-Forms-Technologie realisiert wird. Dessen Hauptaufgabe ist es, dem Anwender die Funktionen zur Erfassung und Verarbeitung von Stapelbuchungen bereitzustellen.

Die Wahl des Entwicklungs-Frameworks für das neue Form war durch die technologische Basis des Financial Suite Cockpits und der Anforderung des SAP Business One SDKs fest vorgegeben. Das SDK, auf dem die gesamte Financial Suite aufbaut, setzt für eine stabile und kompatible Kommunikation mit SAP Business One zwingend das .NET Framework 4.8 voraus.

Ein Einsatz modernerer .NET-Versionen war daher für die Entwicklung des Forms ausgeschlossen. Die Entscheidung für .NET 4.8 war somit keine freie Wahl, sondern eine notwendige technische Voraussetzung, um die Integration in die bestehende Anwendung zu gewährleisten.

Zur Steigerung der Entwicklungseffizienz wurde auf das VerXs Framework zurückgegriffen, eine bereits etablierte Basiskomponente der Financial Suite. Dieses Framework stellt eine Sammlung wiederverwendbarer Funktionen zur Verfügung, die speziell die komplexe Kommunikation mit den SAP-APIs vereinfachen. Die Nutzung dieser erprobten Codebasis verkürzt die Entwicklungszeit für das neue Form erheblich.

#### 4.2 Anforderungsdokument

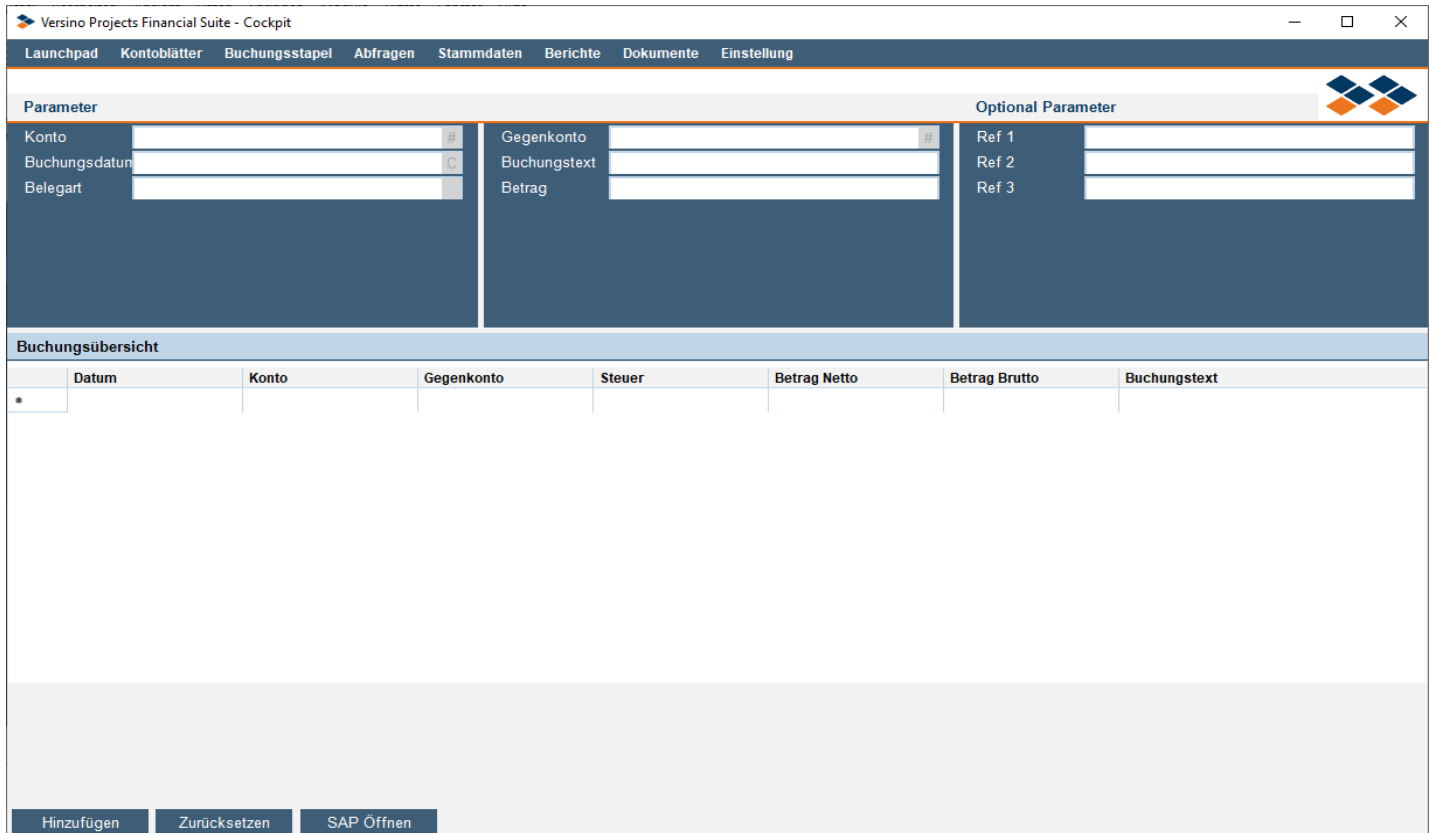
Um den Entwicklungsprozess und die Abrechnung zu vereinfachen, setzen wir kein Lasten- und Pflichtenheft ein. Stattdessen verwenden wir ein sogenanntes Anforderungsdokument, in dem die wesentlichen Anforderungen beschrieben sind. In der Regel bietet dieses Dokument bereits eine visuelle Darstellung der Anforderungen und ihres Aufbaus.

#### 4.3 Visueller Aufbau des Forms

Für die visuelle Gestaltung des Forms wurde eine bereits bestehende Form kopiert und angepasst. Das bot den Vorteil, dass sich das Form später im Aussehen nicht von den anderen Forms unterscheidet. Und dadurch sieht die Financial Suite weiterhin einheitlich aus.

## Implementierung von Buchungstapel in Financial Suite

### Das Form:



The screenshot displays the 'Versino Projects Financial Suite - Cockpit' interface. At the top, a navigation bar includes 'Launchpad', 'Kontoblätter', 'Buchungstapel', 'Abfragen', 'Stammdaten', 'Berichte', 'Dokumente', and 'Einstellung'. The main area is divided into three panels: 'Parameter', 'Optional Parameter', and 'Buchungsübersicht'.

**Parameter Panel:**

Konto	<input type="text"/>	#
Buchungsdatum	<input type="text"/>	C
Belegart	<input type="text"/>	

**Optional Parameter Panel:**

Ref 1	<input type="text"/>
Ref 2	<input type="text"/>
Ref 3	<input type="text"/>

**Buchungsübersicht Panel:**

	Datum	Konto	Gegenkonto	Steuer	Betrag Netto	Betrag Brutto	Buchungstext
*							

At the bottom, there are three buttons: 'Hinzufügen', 'Zurücksetzen', and 'SAP Öffnen'.

Abbildung 1: Buchungstapel-Layout

## 5 Phase 2: Entwicklung der Erweiterung

Dieser Abschnitt beschreibt die technische Realisierung des neuen Forms für das Financial Suite Cockpit. Der Fokus liegt dabei auf zwei Aspekten des neu entwickelten Forms „BookingStackView“. Dessen Benutzeroberfläche sowie die dahinterliegende Programmlogik

### 5.1 Implementierung der Benutzeroberfläche (Form)

Um ein einheitliches Erscheinungsbild innerhalb des gesamten Financial Suite Cockpit zu gewährleisten, wurde die Benutzeroberfläche des Forms „BockingStackView“ so entworfen, dass es sich nahtlos in das bestehende Design einfügt und dem Anwender eine konsistente und intuitive Bedienung ermöglicht. Für den Aufbau kamen spezialisierte Steuerelemente aus dem firmeneigenen VerXs-Framework zum Einsatz. Diese „Item“-Klassen (z.B. ItemTextBox) sind Erweiterungen der Standard Windows Forms Elemente.

Der Obere Bereich enthält drei Panels, die als logischer Container zur Gruppierung der Eingabefelder dienen.



## Implementierung von Buchungstapel in Financial Suite

---

- **MainFilters-Panel:** Erfasst die Grunddaten, die für jede einzelne zu erstellende Transaktion des Stapels herangezogen werden. Dies sind das Hauptkonto (KontoTxt), das Buchungsdatum (BuchDateTxt) und die Belegart (DocTypeBox). Jedes Eingabefeld ist dabei einem Label (z.B. KontoLbl) zugeordnet. Zudem werden die Eingabefelder gesperrt sollten alle befühlt sein.
- **Details-Panel:** Hier werden zu dem im MainFilters-Panel gewählten Hauptkonto die spezifischen Daten für eine einzelne Transaktion vervollständigt: das Gegenkonto (GeKontoTxt), der Buchungstext (BuchTxt) und der Betrag (BetragTxt).
- **Optional Parameter-Panel:** Bietet Felder für optionale Referenzangaben (z.B. Ref1Txt).

Die zentral positionierte Tabelle „Buchungsübersicht“ (ItemDataGrid) zeigt alle erfassten Posten des aktuellen Stapels an. Es bietet dem Anwender eine direkte visuelle Kontrolle über die eingegebenen Daten Pro Buchung.

Wie bereits erwähnt, ist das Form so aufgebaut, dass es sich nahtlos in das bestehende Design des Financial Suite Cockpit einfügt. Aus diesem Grund befinden sich die drei Schaltflächen (Buttons) im unteren linken Bereich Diese haben folgende Funktionen:

- **Hinzufügen:** Initiiert oder ergänzt den internen Buchungstapel (JournalVoucher) um einen neuen Eintrag, der aus den aktuellen Eingabedaten erstellt wird. Anschließend werden diese Daten an die Buchungsübersicht übergeben.
- **Zurücksetzen:** Gibt die Eingabefelder des MainFilters-Panel wieder frei und leert sie.
- **SAP Öffnen:** Schließt den Buchungstapel (JournalVoucher) ab und generiert daraus einen Vorerfassten Beleg, der mittels DI-API an SAP B1 übergeben wird. Unmittelbar danach öffnet sich in SAP B1 das Fenster „Vorerfasste Belege“, wird in den Vordergrund gebracht, und der neu erstellte Beleg ist automatisch markiert.

## 5.2 Implementierung der Programmlogik

Dieser Abschnitt beschreibt die zentrale Programmlogik des Formulars BookingStackView. Im Fokus stehen die Methoden, welche die Verarbeitung der Benutzereingaben steuern und die Interaktion mit dem SAP-System koordinieren.

- **AutolockFields()**
  1. Überprüft die Eingabefelder Konto, Buchungsdatum und Belegart. Wenn alle befühlt sind, werden sie gesperrt.

```
public void AutolockFields(object sender, EventArgs e)
{
    //Check if all fields are filled and lock them
    if (!string.IsNullOrEmpty(KontoTxt.Text) && !string.IsNullOrEmpty(BuchDateTxt.Text) && DocTypBox.SelectedItem != null)
    {
        KontoTxt.Enabled = false;
        BuchDateTxt.Enabled = false;
        DocTypBox.Enabled = false;
    }
}
```

## Implementierung von Buchungstapel in Financial Suite

---

- **UnlockFields()**

1. Gibt die Eingabefelder die durch „AutolockFields()“ gesperrt wurden wieder frei und leert sie.

```
public void UnlockFields(object sender, EventArgs e)
{
    //Unlock the fields that were locked by AutolockFields
    KontoTxt.Enabled = true;
    BuchDateTxt.Enabled = true;
    DocTypBox.Enabled = true;
    //And clear the fields
    KontoTxt.Text = "";
    BuchDateTxt.Text = "";
    DocTypBox.SelectedItem = null;
}
```

- **AddToGrid()**

1. Fügt eine neue Zeile zum BookingGrid hinzu.
2. Mittels SQL-Abfrage werden die Steuerinformationen des Hauptkontos abgefragt

```
public void AddToGrid(object sender, EventArgs e)
{
    //Add new row to the BookingGrid and get its index
    int rowIndex = BookingGrid.Rows.Add();

    //Query to get the default VAT group for the selected account
    var vatGroup = Dyn.Query.GetValues<string>($"SELECT T0."DfltVat" FROM "OACT" T0
    WHERE T0."AcctCode" = '{KontoTxt.Text}').FirstOrDefault();
```

3. Anschließend werden die gesammelten Daten aus den Eingabefeldern und der SQL Abfrage in das BookingGrid verschoben und dort aufgelistet

```
//Fill in the cells with the values from the textboxes
BookingGrid.Rows[rowIndex].Cells["CoKonto"].Value = KontoTxt.Text;
BookingGrid.Rows[rowIndex].Cells["CoDate"].Value = BuchDateTxt.Text;
BookingGrid.Rows[rowIndex].Cells["CoGeKonto"].Value = GeKontoTxt.Text;
BookingGrid.Rows[rowIndex].Cells["CoSteuer"].Value = vatGroup;
BookingGrid.Rows[rowIndex].Cells["CoBuchText"].Value = BuchTxt.Text;
BookingGrid.Rows[rowIndex].Cells["CoRefOne"].Value = RefOneTxt.Text;
BookingGrid.Rows[rowIndex].Cells["CoRefTwo"].Value = RefTwoTxt.Text;
BookingGrid.Rows[rowIndex].Cells["CoRefThree"].Value = RefThreeTxt.Text;
```

## Implementierung von Buchungstapel in Financial Suite

4. Nun wird versucht den Inhalt aus „BetragTxt“ in eine Decimalzahl umzuwandeln. Sollte dies erfolgreich sein wird sie ebenfalls an das BookingGrid übergeben.
5. Anschließend wird damit, und den Steuerinformationen, der Nettobetrag ermittelt.

```
//Try to parse the amount entered by the user
if (decimal.TryParse(BetragTxt.Text, NumberStyles.Any, info, out decimal betrag))
{
    //Set the gross amount (Brutto) in the grid
    BookingGrid.Rows[rowIndex].Cells["CoBeBrutto"].Value = betrag.ToString("N2", info);
    ...
    double vatRate = 0;
    if (!string.IsNullOrEmpty(vatGroup))
    {
        //If a VAT group is found, get the VAT rate and account
        var vatData = Dyn.Query.GetValues<(string rate, string account)>($"@SELECT T0."Rate",T0."Account" FROM
        ... "OVTG" T0 WHERE T0."Code" = '{vatGroup}').FirstOrDefault();
        double.TryParse(vatData.rate, NumberStyles.Any, CultureInfo.InvariantCulture, out vatRate);
    }
    ...
    decimal netto = betrag;
    if (vatRate > 0)
    {
        //Calculate the amount (Netto) if VAT is applicable
        netto = betrag / (decimal)(1 + (vatRate / 100));
    }
    //Set the net amount (Netto) in the grid
    BookingGrid.Rows[rowIndex].Cells["CoBeNetto"].Value = netto.ToString("N2", info);
}
```

- **CreateJournalVoucher()**

1. Greift auf klassenweite Variable JournalVoucher zu und instanziiert bei Bedarf ein neues JournalVoucher-Objekt der DI-API.
2. List die Daten aus der zuletzt zum BookingGrid hinzugefügten Zeile aus.
3. Speichert die Daten aus dem BookingGrid in Lokalen Variablen zur weiteren Verarbeitung.

```
// Class-level variable that holds a reusable journal voucher instance for posting entries.
private JournalVouchers? JournalVoucher;
1 Verweis
public void CreateJournalVoucher(object sender, EventArgs e)
{
    //Check if the grid has at least one non-empty row
    if (BookingGrid.Rows.Count > 0 && !BookingGrid.Rows[0].IsNewRow)
    {
        //Get the index of the last valid row (excluding the new row placeholder)
        int lastRowIndex = BookingGrid.Rows.Count - 2;
        if (lastRowIndex < 0)
        {
            //No valid row found
            MessageBox.Show("Keine gültige Zeile gefunden.", "Fehler", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
        //Retrieve the last row from the grid
        DataGridViewRow row = BookingGrid.Rows[lastRowIndex];

        //Extract values from the row's cells
        string bdatetxt = row.Cells["CoDate"].Value.ToString();
        string konto = row.Cells["CoKonto"].Value.ToString();
        string gkonto = row.Cells["CoGeKonto"].Value.ToString();
        string steuer = row.Cells["CoSteuer"].Value.ToString();
        string buchtxt = row.Cells["CoBuchText"].Value.ToString();
        double betrag = double.Parse(row.Cells["CoBeBrutto"].Value?.ToString() ?? "0", NumberStyles.Any, info);
        double netto = double.Parse(row.Cells["CoBeNetto"].Value?.ToString() ?? "0", NumberStyles.Any, info);
        string ref1 = row.Cells["CoRefOne"].Value.ToString();
        string ref2 = row.Cells["CoRefTwo"].Value.ToString();
        string ref3 = row.Cells["CoRefThree"].Value.ToString();
    }
}
```

## Implementierung von Buchungsstapel in Financial Suite

4. Nun wird mittels SQL-Abfrage der Kontotype abgefragt. Dies ist wichtig für die Zuordnung von Soll und Haben in der Buchung

```
//Query the account type (e.g., Income, Expense, Other)
var kontoTyp = Dyn.Query.GetValues<string>($"
SELECT T0."AcctType" FROM "OACT" T0 WHERE T0."AcctCode" = '{konto}'.FirstOrDefault();
```

5. Nun wird überprüft ob „konto“, „betrag“ und „netto“ über gültige Daten verfügen.
6. Sollte das so sein, wird ein neues JournalVoucher-Objekt erstellt.
7. Daraufhin werden die ersten Daten wie „Datum“, „Buchungstext“ und „Ref“ übergeben.

```
//Checks if account is valid and amount is greater than zero
if (!string.IsNullOrEmpty(konto) && (betrag > 0 || netto > 0))
{
    //Create or reuse the JournalVoucher object
    if (JournalVoucher == null)
    {
        JournalVoucher = (JournalVouchers)Dyn.DiCompany.GetBusinessObject(BoObjectTypes.oJournalVouchers);
    }
    //Set general properties of the journal voucher
    JournalVoucher.JournalEntries.ReferenceDate = DateTime.Parse(bdatetxt, info);
    JournalVoucher.JournalEntries.DueDate = DateTime.Parse(bdatetxt, info);
    JournalVoucher.JournalEntries.TaxDate = DateTime.Parse(bdatetxt, info);
    JournalVoucher.JournalEntries.Memo = buchtxt;
    JournalVoucher.JournalEntries.Reference = ref1;
    JournalVoucher.JournalEntries.Reference2 = ref2;
    JournalVoucher.JournalEntries.Reference3 = ref3;
```

8. Im Folgenden wird mittels if-Abfrage kontrolliert, ob „steuer“ einen Wert hat.
9. Sollte „steuer“ einen Wert haben wird diese auch berechnet, andernfalls nicht.
10. Mit Hilfe des ermittelten „kontoTyp“ wird entschieden wie Soll und Haben gebucht wird.

```
//Enable or disable automatic VAT calculation based on the presence of a tax code
if (!string.IsNullOrEmpty(steuer))
{
    JournalVoucher.JournalEntries.AutoVAT = BoYesNoEnum.tYES;
}
else
{
    JournalVoucher.JournalEntries.AutoVAT = BoYesNoEnum.tNO;
}

//Add journal lines based on the account type
if (kontoTyp == "I")
{
    JournalVoucher.JournalEntries.Lines.AccountCode = konto;
    JournalVoucher.JournalEntries.Lines.Credit = netto;
    JournalVoucher.JournalEntries.Lines.TaxCode = steuer;
    JournalVoucher.JournalEntries.Lines.Add();

    JournalVoucher.JournalEntries.Lines.AccountCode = gkonto;
    JournalVoucher.JournalEntries.Lines.Debit = betrag;
    JournalVoucher.JournalEntries.Lines.Add();
}
else if (kontoTyp == "E")
{
    JournalVoucher.JournalEntries.Lines.AccountCode = konto;
    JournalVoucher.JournalEntries.Lines.Debit = netto;
    JournalVoucher.JournalEntries.Lines.TaxCode = steuer;
    JournalVoucher.JournalEntries.Lines.Add();

    JournalVoucher.JournalEntries.Lines.AccountCode = gkonto;
    JournalVoucher.JournalEntries.Lines.Credit = betrag;
    JournalVoucher.JournalEntries.Lines.Add();
}
```

## Implementierung von Buchungsstapel in Financial Suite

---

11. Abschließend wird der Eintrag dem JournalVoucher hinzugefügt.

```
// Add the journal entry to the JournalVoucher
int result = JournalVoucher.JournalEntries.Add();
if (result != 0)
{
    string errorMessage = Dyn.DiCompany.GetLastErrorDescription();
    MessageBox.Show($"Fehler beim Erstellen des Buchungsstapels: {errorMessage}", "Fehler", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

- **VoucherSendToSAP()**

1. In dieser Methode wird der Buchungsstapel (JournalVoucher) geschlossen und an SAP B1 übergeben.

```
public void VoucherSendToSAP(object sender, EventArgs e)
{
    //Check if the JournalVoucher object exists has entries
    if (JournalVoucher != null)
    {
        //Send the JournalVoucher to SAP
        JournalVoucher.Add();
        MessageBox.Show("Voucher erfolgreich erstellt");
    }
}
```

## 6 Phase 3 Testen und Anpassen

### 6.1 Funktionale und Logische Prüfung

Nach dem Abschluss der Implementierungsphase folgte nun eine grundlegende Prüfung aller implementierten Funktionen. In dem Bestreben, potenzielle Fehlerquellen zu identifizieren, habe ich mich bemüht, durch das Testen verschiedener Konstellationen gezielt Schwachstellen zu finden und zu beheben.

Bei detaillierten Tests offenbarte sich eine funktionale Einschränkung. Der soeben erstellte Vorerfasste Beleg (JournalVoucher) konnte nicht für die direkte Weiterverarbeitung zu Journalbuchungen abgerufen werden. Da die Lösung dieser Problematik einen erheblichen Mehraufwand bedeutet hätte, der den zeitlichen Rahmen des Projektes überschritten hätte, wurde beschlossen eine einfachere Lösung zu nutzen.

Die als Alternative gewählte Lösung wurde als fester Bestandteil in den Ablauf der Methode „VoucherSendToSAP()“ integriert. Dies hat den Vorteil, dass die Funktionalität dieser Lösung nun bei jedem Aufrufen der „VoucherSendToSAP()“ Methode automatisch und unmittelbar mit ausgeführt wird.

## Implementierung von Buchungstapel in Financial Suite

Der folgende Code-Abschnitt dient dazu, den Anwender umgehend zu SAP Business One weiterzuleiten und das Fenster „Vorerfasste Belege“ zu öffnen.

Dadurch das „Vorerfasste Belege“ geöffnet wird, fragt der Code-Abschnitt die FormID ab.

Sollte die FormID von „Vorerfasste Belege“ gleich „229“ sein wird der Inhalt von if ausgeführt.

Der Inhalt des if sorgt dafür das der eben erstellte JournalVoucher sofort markiert wird, so dass der Anwender nur noch auf den Button „Mappe buchen“ drücken muss.

Zudem werden alle Zeilen des BookingGrids gelöscht.

```
//Open the SAP business one menu for Journal Vouchers
B1LinkHandle.OpenB1Menu("1541");

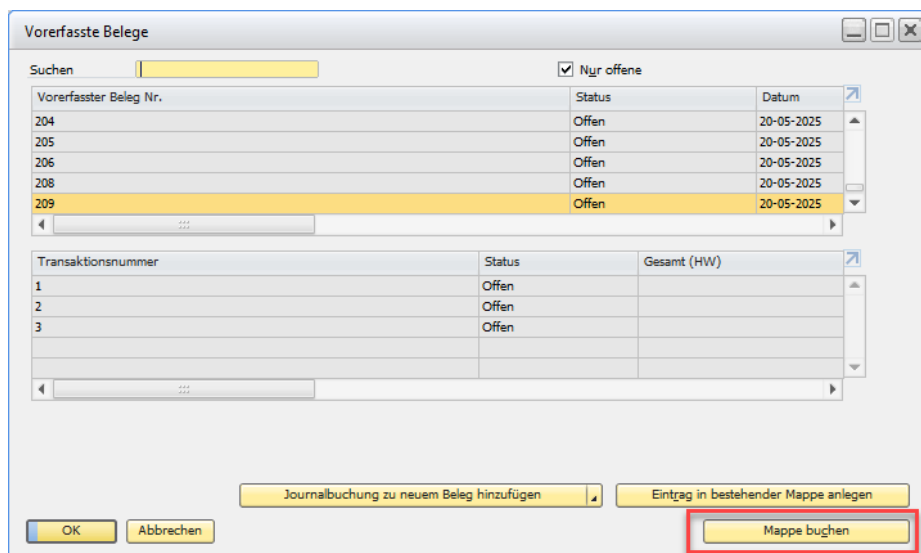
//Get the currently active SAP form
SAPbouiCOM.Form form = Dyn.UiApplication.Forms.ActiveForm;

//Check if the active form is the Journal Vouchers form "229"
if (form.TypeEx == "229")
{
    //Get the matrix control from the form
    Matrix matrix = (Matrix)form.Items.Item("8").Specific;

    //Get the number of rows in the matrix
    int letzteZeile = matrix.RowCount;

    //If there are rows in the matrix, select the last row
    if (letzteZeile >= 0)
    {
        matrix.Columns.Item("1").Cells.Item(letzteZeile).Click(BoCellClickType.ct_Regular);
    }
}

//Clear the BookingGrid for new entries
BookingGrid.Rows.Clear();
```



Vorerfasster Beleg Nr.	Status	Datum
204	Offen	20-05-2025
205	Offen	20-05-2025
206	Offen	20-05-2025
208	Offen	20-05-2025
209	Offen	20-05-2025

Transaktionsnummer	Status	Gesamt (HW)
1	Offen	
2	Offen	
3	Offen	



## Implementierung von Buchungstapel in Financial Suite

---

## 7 Produktdokumentation

### 7.1 Anwenderdokumentation

Um den Umfang der Funktionen und deren Anwendungen anschaulicher zu gestalten, wurde nach Abschluss der Testphase eine Anwendungsdokumentation für die zukünftigen Anwender erstellt. (Anhang A.3)

### 7.2 Entwicklerinformation

Neben der externen Projektdokumentation wurde großer Wert auf eine 'interne' Dokumentation direkt im Quellcode gelegt. Dies umfasste zum einen die konsequente Verwendung sprechender Namen für Variablen, Methoden und Klassen, um deren Zweck unmittelbar verständlich zu machen. Zum anderen wurden relevante Code-Abschnitte mit Kommentaren versehen, die die implementierte Logik und die Absichten dahinter erläutern. Diese Praktiken sind fundamental für die Sicherstellung der Wartbarkeit und erlauben zukünftigen Entwicklern eine effiziente Einarbeitung sowie eine problemlose Weiterentwicklung

## 8 Fazit

### 8.1 Soll-/Ist- Vergleich

Nach einer umfassenden Reflexion über die nahezu abgeschlossene IHK-Projektarbeit ergibt sich, dass nicht alle zuvor definierten Anforderungen ohne Komplikationen bewältigt werden konnten. Die in Abbildung 2 dargestellte Soll/Ist-Zeitplan-Tabelle veranschaulicht, dass für einige Phasen weniger, jedoch für andere wesentlich mehr Zeit in Anspruch genommen werden musste.

Die Abweichung vom ursprünglichen Zeitplan ist der technischen Problematik bei der Handhabung des JournalVoucher (Vorerfassten Beleg) geschuldet. Konkret war ein direkter Wiederaufruf zur Weiterverarbeitung nach dessen Erstellung nicht möglich. Dieser Umstand machte Anpassungen in der Testphase und der Dokumentation notwendig, was die Zeitplanung laut Abbildung 2 beeinflusste.

Phasen des Projekts	Soll	Ist
Phase 1: Analyse und Anforderungsdefinition	15 h	12 h
Phase 2: Entwicklung der Erweiterung	35 h	30 h
Phase 3: Testen und Anpassungen	20 h	23 h
Phase 4: Erstellung der Dokumentation	10 h	20 h
<b>Gesamtzeit</b>	<b>80 h</b>	<b>85 h</b>

Abbildung 2: Soll-/Ist- Zeitplan Tabelle

## Implementierung von Buchungstapel in Financial Suite

---

### 8.2 Gesammelte Erfahrungen

Während der Durchführung dieses Projekts erlangte ich nicht nur wichtige Erfahrungen in der Planung und Ausführung von Projekten, sondern erhielt auch wertvolle Einblicke in die Arbeitsweise von SAP Business One.

Über den Verlauf des Projekts wurde mir klar, wie wichtig eine sorgfältige Planung ist und welchen Mehrwert eine strukturierte Herangehensweise mit sich bringt. Insbesondere die Erfahrung mit unvorhergesehenen technischen Herausforderungen, wie das Problem beim direkten Wiederaufruf des JournalVoucher, verdeutlichte, wie schnell Abweichungen vom ursprünglichen Plan entstehen können. Diese Erkenntnisse haben meine Sichtweise auf Projektmanagement und -ausführung erweitert und mich für zukünftige Vorhaben besser vorbereitet.

Darüber hinaus ermöglicht mir die Arbeit an diesem Projekt, mich intensiver mit den genutzten Frameworks, insbesondere VerXs und SAP Business One, auseinanderzusetzen. Diese tiefergehenden Kenntnisse werden es mir ermöglichen, bei künftigen Aufgaben eine effizientere und gezieltere Arbeitsweise an den Tag zu legen.

Insgesamt betrachtet war die Durchführung dieses Projekts nicht nur eine Bereicherung für das Unternehmen, sondern auch für mich persönlich. Ich gewann wertvolle Eindrücke und Fähigkeiten, insbesondere im Umgang mit SAP Business One, die einen positiven Beitrag zu meiner beruflichen Entwicklung leisteten.

### 8.3 Ausblick für zukünftige Anpassungen

Zwar wurden im Zuge des Projekts nicht alle genannten Anforderungen erfüllt, doch wurde schon über mögliche Erweiterungen nachgedacht. Zum Beispiel durch das Einfügen von anderen Belegarten wie Ausgangsrechnung und Eingangsrechnung.

Eine Erweiterung mit diesen Belegarten würde es ermöglichen, dass das System nach der Eingabe grundlegender Daten wie Konto/Geschäftspartner, Datum, Betrag und der gewählten Belegart (z.B. Ausgangsrechnungen) proaktiv nach offenen Rechnungen sucht, die diesen Kriterien entsprechen. Dem Anwender könnten dann potenziell passende Belege vorgeschlagen werden, die er zur direkten Verbuchung auswählen kann, was die manuelle Zuordnung erheblich vereinfachen würde.

Zudem würden die gebuchten Belege auch in dem dafür angepassten BookingGrid angezeigt werden, was bietet den Vorteil dass der Anwender nicht den Überblick über bereits gebuchte Rechnungen verliert.



## Implementierung von Buchungsstapel in Financial Suite

### A Anhang

#### A.1 Zeitliche Planung des Projekts

<b>Analyse und Anforderungsdefinition</b>	<b>15 h</b>
Analyse der bestehenden Systeme und Prozesse	5 h
Klärung von Sonderfällen und Ausnahmen	5 h
Erstellung und Abstimmung des Anforderungsdokuments	5 h
<b>Entwicklung der Erweiterung</b>	<b>35 h</b>
Design und Architektur der Erweiterung	5 h
Programmierung der Erweiterung	20 h
Integration und Code-Reviews	10 h
<b>Testen und Anpassungen</b>	<b>20 h</b>
Erstellung von Testfällen und Durchführung von Tests	10 h
Identifikation und Behebung von Fehlern	7 h
Anpassung der Erweiterung basierend auf Testergebnissen	3 h
<b>Erstellung der Dokumentation</b>	<b>10 h</b>
Erstellung von Benutzerhandbuch und technischer Dokumentationen	5 h
Dokumentation der Implementierung und Konfiguration	3 h
Überprüfung und Freigabe der Dokumentation	2 h
<b>Gesamtzeit</b>	<b>80 h</b>

Abbildung 3: Zeitplan Tabelle

#### A.2 Verwendete Ressourcen

##### Software

- Windows 10 Professional – Betriebssystem Entwicklung Test und Tests
- Microsoft Edge – Browser
- Visual Studio 2022 Community – Entwicklungsumgebung
- C# - Programmiersprache der Anwendung
- SQL – Programmiersprache bei Datenbank Kommunikation
- .NET4.8 – Framework Microsoft
- VerXs Framework – Framework zur Erweiterung für SAP B1 SDK
- MSSQL-Server 2019 – Verwendete Datenbank
- SAP B1 Client Installation – Ziel Anwendung
- SAP B1 Server Installation – Ziel Server
- Office 365 – Verwendete Werkzeuge zur Anfertigung der Dokumentation

##### Hardware

- Büroarbeitsplatz
- Notebook mit vorinstallierter Entwicklungsumgebung
- Firmenserver zur Bereitstellung von virtualisierter Test-Umgebung

## Implementierung von Buchungsstapel in Financial Suite

### A.3 Anwenderdokumentation



#### Buchungsstapel für Financial Suite

##### Anwenderdokumentation: Buchungsstapel für Financial Suite

###### Herzlich willkommen!

Schön, dass Sie sich für das Modul "Buchungsstapel" unserer Financial Suite interessieren! Diese Dokumentation begleitet Sie Schritt für Schritt durch die Funktionen und zeigt Ihnen, wie Sie damit ähnliche oder wiederkehrende Buchungen deutlich schneller erfassen können. Ziel ist es, Ihnen wertvolle Zeit im Buchhaltungsalltag zu sparen und Ihre Arbeitsprozesse weiter zu optimieren.

#### 1. Was Sie benötigen: Systemvoraussetzungen

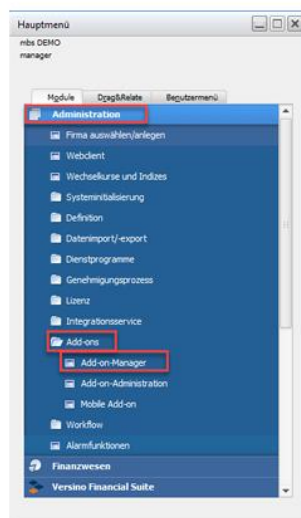
Damit Sie das Modul "Buchungsstapel" optimal nutzen können, sind folgende Systemvoraussetzungen notwendig:

- Ein funktionierendes **SAP Business One System**.
- Die **Financial Suite** muss mitsamt dem **Financial Cockpit** auf Ihrem System installiert sein. Wichtig zu wissen: Die Installation der Financial Suite übernehmen autorisierte Mitarbeiter Ihres IT-Supports oder unseres Serviceteams.

#### 2. So starten Sie das Modul "Buchungsstapel"

Der Weg zum Buchungsstapel ist einfach. Folgen Sie diesen Schritten:

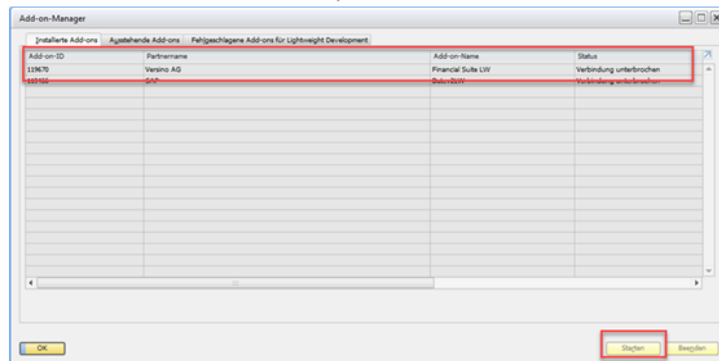
1. Öffnen Sie **SAP Business One** und melden Sie sich wie gewohnt an.
2. Im Hauptmenü finden Sie den Pfad: **Administration > Add-ons > Add-on-Manager**.



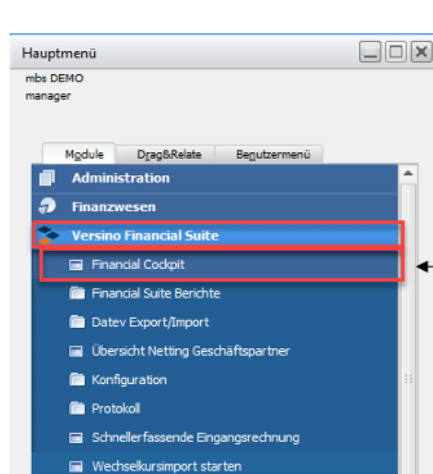
## Implementierung von Buchungstapel in Financial Suite

### Buchungstapel für Financial Suite

3. In der Liste der Add-ons suchen Sie die **"Versino Financial Suite"** heraus und klicken dann auf „**Starten**“. (Dies ist jedoch nur erforderlich, wenn das Add-on nicht im Autostart ist)

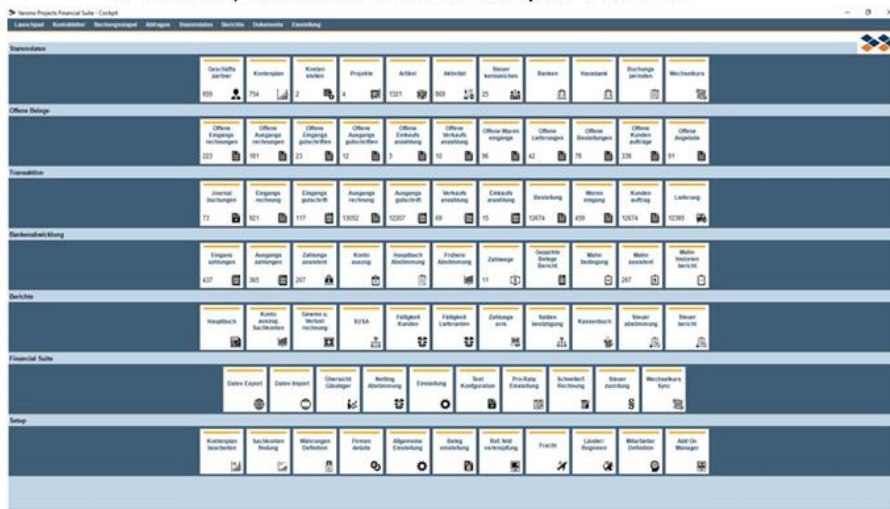


4. Ist das Add-on gestartet, erscheint im Hauptmenü von SAP Business One ein neuer Reiter: **"Versino Financial Suite"**.
5. Klicken Sie auf diesen Reiter und wählen Sie dort **"Financial Cockpit"**.



## Implementierung von Buchungsstapel in Financial Suite

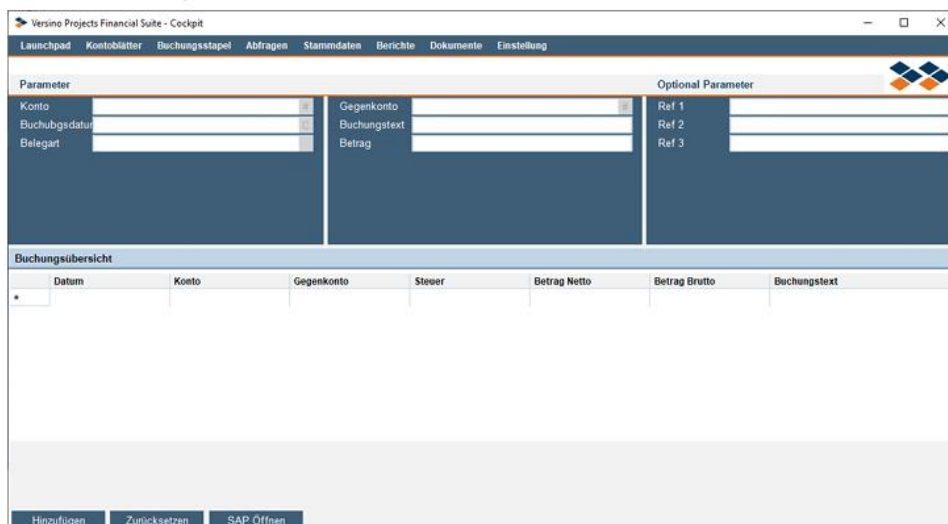
6. Das Hauptfenster des "Financial Cockpits" öffnet sich.



7. Im oberen Bereich des Financial Cockpits sehen Sie eine Menüleiste. Klicken Sie dort auf die Schaltfläche „**Buchungsstapel**“.



8. Perfekt! Die Benutzeroberfläche des Moduls "Buchungsstapel" ist nun vor Ihnen.



## Implementierung von Buchungsstapel in Financial Suite

---



### Buchungsstapel für Financial Suite

### 3. Die Arbeit mit dem Buchungsstapel

Sie haben das Modul "Buchungsstapel" geöffnet – sehr gut! Die Oberfläche ist darauf ausgelegt, Ihnen die Vorerfassung, Bearbeitung und schlussendliche Buchung Ihrer Belege so einfach wie möglich zu machen.

#### 3.1. Einen neuen vorefassten Beleg anlegen

Ein neuer vorefasster Beleg ist schnell angelegt. Beginnen Sie mit diesen initialen Feldern. Sobald diese drei korrekt ausgefüllt sind, werden sie gesperrt. Das sichert die Datenkonsistenz für die folgenden Buchungspositionen. Möchten Sie die Eingaben ändern oder leeren, nutzen Sie einfach den „Zurücksetzen“-Button.

- **1. Konto:**
  - Hier wählen Sie das Sach- oder Geschäftspartnerkonto.
  - Ein Klick auf das graue „#“-Symbol rechts im Feld öffnet die bekannte Kontenauswahl aus SAP Business One.
- **2. Buchungsdatum:**
  - Auch hier hilft ein Klick auf die graue Schaltfläche, um ein Datum aus dem Kalender auszuwählen.
  - Ein kleiner Kniff: Tippen Sie ein beliebiges Zeichen ein und drücken Sie die **Tab-Taste** – schon steht das aktuelle Tagesdatum im Feld.
- **3. Belegart:**
  - Legen Sie fest, welche Art von Beleg Sie erfassen.
  - (Aktuell unterstützt das Modul die Erfassung von Journalbuchungen.)

Sind diese drei Felder gefüllt, sind sie vorerst fixiert. Für Korrekturen steht Ihnen, wie erwähnt, der „Zurücksetzen“-Button zur Verfügung.

- **4. Gegenkonto:**
  - Die Auswahl funktioniert genauso wie beim Feld Konto. Nutzen Sie das „#“-Symbol für die Kontoauswahl.
  - Das Gegenkonto ist das Konto was dem gewählten Hauptkonto gegenübergestellt wird.
- **5. Buchungstext:**
  - Formulieren Sie hier einen klaren Text, der den Geschäftsvorfall gut beschreibt. Dieser Text wird später auch in den Journalbuchungen angezeigt.

## Implementierung von Buchungsstapel in Financial Suite

---



### Buchungsstapel für Financial Suite

- **6. Betrag:**
  - Geben Sie hier den Bruttobetrag für die Buchungsposition an.
  - Keine Sorge um das Netto und Steuer: Der Nettobetrag sowie die Steueranteile werden vom System automatisch berechnet, basierend auf den hinterlegten Steuerinformationen.
- **7. Ref 1 / Ref 2 / Ref 3 (Optional):**
  - Diese Felder stehen Ihnen für zusätzliche, optionale Referenzangaben zu jeder Buchungsposition zur Verfügung. "Ref" steht dabei für "Referenz". Nutzen Sie diese Felder, um beispielsweise interne Vermerke oder spezifische Kennzeichnungen zu hinterlegen.

### 3.2. Funktion der Buttons

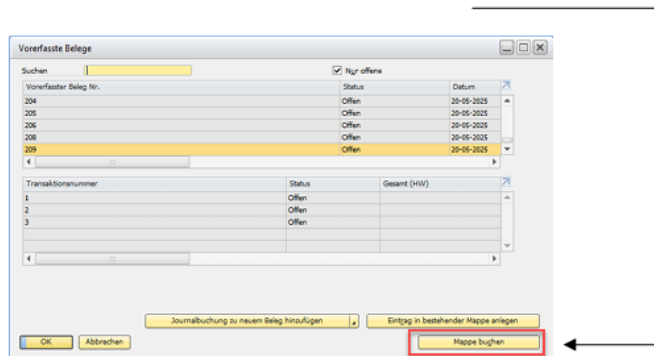
Mit den Buttons am unteren Rand steuern Sie den gesamten Prozess von der Erfassung einzelner Positionen bis zur Übergabe des fertigen Stapels. Sie ermöglichen das Hinzufügen von Einträgen, das Zurücksetzen von Eingabefelder und schließlich das Übertragen der Buchungen an SAP Business One.

- **Hinzufügen:**
  - Übernimmt die aktuell erfasste Buchungsposition komfortabel und sicher in Ihren Buchungsstapel, sobald Sie alle relevanten Positionsdetails vollständig eingegeben haben.
  - Ihre Eingabe erscheint als neue Zeile in der "Buchungsübersicht", wodurch Ihr Stapel wächst, und die Eingabefelder sind danach meist für den nächsten Eintrag bereit.
- **Zurücksetzen:**
  - Dieser Button ermöglicht es Ihnen, wie bereits in Punkt 3.1 angegeben, die Felder „Konto“, „Buchungsdatum“ und „Belegart“ wieder freizugeben und zu leeren.

## Implementierung von Buchungsstapel in Financial Suite

- **SAP Öffnen:**

- Mit diesem Button erstellen Sie final den Vorerfassten Beleg mit all seinen Unterbelegen.
- Danach öffnet sich automatisch das Fenster „Vorerfasste Belege“ in SAP Business One und holt es in den Vordergrund. Zudem wird der von ihnen erstellte Beleg markiert, alles, was Sie noch tun müssen, ist auf „Mappe buchen“ drücken.



Herzlichen Glückwunsch! Sie haben nun das notwendige Wissen an der Hand, um das Modul 'Buchungsstapel' souverän zu bedienen und dessen Vorteile voll auszuschöpfen. Die Fähigkeit, Buchungen effizient im Stapel zu verarbeiten, wird Ihren Arbeitsalltag positiv verändern

**Wichtig:** Sollten Sie das Cockpit aus irgendeinem Grund schließen, aber davor nicht auf „SAP Öffnen“ gedrückt haben, gehen leider alle bisher eingespielten Buchungen verloren.