

České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská

Katedra matematiky
Obor: Aplikovaná informatika



Programové vybavení drobného podnikatele

Software solution for small businesses

BACHELOR'S DEGREE PROJECT

Vypracovala: Nela Koliášová
Vedoucí práce: Doc. Ing. Miroslav Virius, CSc.
Rok: 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Nela Koliášová
Studijní program: Aplikace přírodních věd
Studijní obor: Aplikovaná informatika
Název práce (česky): Programové vybavení drobného podnikatele
Název práce (anglicky): Software solution for small businesses

Pokyny pro vypracování:

- 1) Seznamte se s problematikou vedení účetnictví a další evidence v malé firmě.
- 2) Sestavte požadavky na softwarové vybavení vyhovující zjištěným požadavkům.
- 3) Analyzujte tyto požadavky a navrhnete na základě této analýzy desktopovou aplikaci pro drobného podnikatele.
- 4) Navrženou aplikaci implementujte.
- 5) Otestujte vyvinutou aplikaci a sestavte k ní uživatelskou příručku.

Doporučená literatura:

- 1) Zákon o účetnictví (úplné znění). Předpis č. 563/1991 Sb. ve znění platném od 1. 1. 2021. Dostupné na <https://www.mesec.cz/zakony/zakon-o-ucetnictvi/uplne/>
- 2) M. Virius, Programování v C#. Od základů k profesionálnímu použití. Grada Publishing, Praha, 2021.
- 3) D. M. Kroenke, Databáze. Albatros Media, Praha, 2015.

Jméno a pracoviště vedoucího bakalářské práce:

Doc. Ing. Miroslav Virius, CSc.
KSI FJFI ČVUT v Praze, Trojanova 13, 120 00 Praha 2

Jméno a pracoviště konzultanta:

Datum zadání bakalářské práce: 31.10.2021

Datum odevzdání bakalářské práce: 7.7.2022

Doba platnosti zadání je dva roky od data zadání.

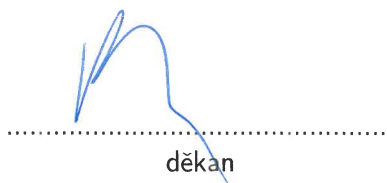
V Praze dne 20. května 2022



.....
garant oboru



.....
vedoucí katedry


.....
děkan

Declaration

I declare that this Bachelor's Degree Project is entirely my own work and I have listed all the used sources in the bibliography.

Prague, July 7, 2022

.....
Nela Koliášová

Acknowledgements

I would like to thank Doc. Ing. Miroslav Virius, CSc. for his expert guidance and express my gratitude to Mgr. Hana Čápová for her language assistance.

Nela Koliášová

Název práce:

Programové vybavení drobného podnikatele

Autor: Nela Koliášová

Studijní program: Aplikace přírodních věd

Obor: Aplikovaná informatika

Druh práce: Bachelor's Degree Project

Vedoucí práce: Doc. Ing. Miroslav Virius, CSc.

KSI FJFI ČVUT v Praze, Trojanova 13, 120 00 Praha 2

Abstrakt: Vedení daňové evidence je poměrně jednoduchý proces. Aplikace, které k němu slouží, ale bývají složité a drahé. Pokud takové problémy nemají, není v nich na druhou stranu možné evidovat majetek, ale pouze vést evidenci příjmů a výdajů. Tato bakalářská práce popisuje proces návrhu a implementace aplikace, která umožňuje evidovat vše, co je ze zákona povinné, a to v uživatelsky přívětivém prostředí. Analyzován je i samotný proces vedení daňové evidence. V práci je také zhodnocena nová knihovna na tvorbu uživatelského prostředí, WinUI 3, pomocí které bylo vytvořeno uživatelské prostředí aplikace, a jsou představeny některé problémy této knihovny, které zatím nebyly vyřešeny.

Klíčová slova: daňová evidence, databázová aplikace, C#, WinUI 3, Entity framework

Title:

Software solution for small businesses

Author: Nela Koliášová

Abstract: Keeping tax records is a fairly simple process. Applications for it are, however, usually complicated or expensive. In other cases, they only cover the process of keeping income and expense records, but not asset records. This Bachelor's Degree Project describes the process of designing and implementing an application that would not only cover the whole process, but also do so in a user-friendly way. The process of keeping tax records is also analyzed. In addition, the new user interface framework, WinUI 3, which was used to create the application, is evaluated and some problems of this framework that have not yet been solved are introduced.

Key words: tax records, database application, C#, WinUI 3, Entity framework

Contents

Introduction	1
1 Theory	3
1.1 Terms used	3
1.2 Small businessman	3
1.3 Tax records	3
1.3.1 Income and expense records	4
1.3.2 Asset records	4
1.3.3 Claim and debt records	7
1.3.4 Income tax	7
1.4 Flat tax	7
1.5 Bookkeeping	8
2 Analysis	9
2.1 Consultation with a field expert	9
2.2 Competitive analysis	9
2.2.1 Excel	10
2.2.2 Money S3	10
2.2.3 Účto	10
2.2.4 iDoklad	11
2.2.5 Fakturoid	11
2.2.6 Summary	11
2.3 Requirements	12
2.3.1 Functional requirements	12
2.3.2 Non-functional requirements	12
2.4 Actors	12
2.5 Use cases	13
2.5.1 UC1: Adding a new item for income and expense records	13
2.5.2 UC2: Adding a new card for asset records	13
2.5.3 UC3: Automatically depreciating assets and adding it to expenses	14
2.5.4 UC4: Adding a new item for claim and debt records	14
2.5.5 UC5: Accessing income and expense list	14
2.5.6 UC6: Accessing asset list	15
2.5.7 UC7: Accessing asset card detail	15
2.5.8 UC8: Accessing claim and debt list	15
2.5.9 UC9: Marking a payment of a claim or a debt	15
2.5.10 UC10: Editing records	16
2.5.11 UC11: Deleting records	17
2.5.12 UC12: Filtering records	17
2.5.13 UC13: Sorting records	17
2.5.14 UC14: Displaying income and expense overview	18
2.5.15 UC15: Displaying necessary data for filling in a tax return form	18

2.5.16	Use case diagram	18
2.6	Requirement coverage	20
3	Technologies and design	21
3.1	Technologies	21
3.1.1	Windows App SDK	21
3.1.2	C#	22
3.1.3	SQLite	22
3.1.4	Entity framework	22
3.1.5	NuGet Packages	22
3.1.6	Single-project MSIX Packaging Tools	23
3.2	Architecture	23
3.2.1	Graphical User Interface	23
3.2.2	Application logic	25
3.2.3	Database connection	26
3.2.4	Database	26
3.2.5	Architecture diagram	28
4	Realization	29
4.1	Class diagram	30
4.2	Classes and methods	31
4.2.1	Main window	31
4.2.2	Classes representing pages	31
4.2.3	Communication with database	33
4.2.4	Application logic class	34
4.2.5	App class	34
4.3	Publishing	34
4.4	Problems with WinUI 3 discovered	35
5	User acceptance testing	37
5.1	Testers	37
5.2	Process	38
5.2.1	Briefing	38
5.2.2	Scenario	38
5.2.3	Debriefing	39
5.3	Results	39
5.3.1	Person A's comments	39
5.3.2	Person B's comments	40
5.3.3	Person C's comments	40
5.3.4	Person D's comments	41
5.3.5	Summary	41
5.4	User manual	42
	Conclusion	43
	Bibliography	45
	Attachments	47
A	Contents of the enclosed CD	47
B	Glossary	48
C	Uživatelská příručka	49
C.1	Podporovaná zařízení	49
C.2	Přidání příjmu nebo výdaje	49

C.3	Přidání pohledávky nebo dluhu	49
C.4	Přidání majetku	49
C.5	Práce s tabulkami	49
C.6	Deník příjmů a výdajů	50
C.7	Pohledávky a dluhy	50
C.8	Karty majetku	50
C.9	Drobný majetek	51
C.10	Mazání a editace záznamů	51
C.11	Filtry a hledání	52
C.12	Odpisy majetku	53
C.13	Stránka Přehled	53
C.14	Stránka Nápověda	53
D	Účto	55
E	Screenshots from the application	61
F	Directory tree	78

Introduction

The process of keeping tax records in the Czech Republic was designed to be as simple as possible. The law is not strict about any rules regarding it, and the rules it does give are quite elementary. Despite this, applications created for this process are very often over-complicated or intimidating to people with no accounting background. Even if there is a solution that can be navigated with ease, it is usually quite expensive. However, there is a considerable number of small businessmen with relatively low earnings, who try to keep the tax records at minimal cost. For all of these reasons, small businessmen often opt for some spreadsheet software instead, but, using it, they have no control over the accuracy of their calculations and can accidentally lose valuable data.

The main aim of this Bachelor's Degree Project is to create a desktop application that would eliminate all of the issues described above. Its UX (user experience) should be intuitive and there should be an emphasis on a modern and familiar design. If possible, it should be able to provide explanations of some terms for people who might struggle with the terminology. The application should, in addition, be then provided for free to small businessmen to use, ideally on a widely used digital distribution service, like the Microsoft Store.

Microsoft has recently introduced a new framework WinUI 3, which allows the developer to use the same components as Windows 11. This application is a great opportunity to utilize this new framework and to discover what kind of application it can create.

This Bachelor's Degree Project describes the process of creating such an application. It consists of five chapters. In the first chapter, the issue of income tax is described, as well as the theory about the process of keeping tax records. Chapter two, analysis, discusses competition and sets a clear set of requirements and use cases. The third chapter goes into detail about technologies to be used and the presumed architecture of the program. The realization chapter will describe the process of programming and the structure of the final program. Finally, the last chapter, testing, will summarize the results of the user testing, describe the process and introduce any eventual changes made in the application after this stage. For creating diagrams, the *draw.io* online tool was used.

Chapter 1

Theory

1.1 Terms used

This thesis contains numerous terms which can have different meanings in the English and Czech languages. Moreover, there are several Czech laws and acts that have neither direct translation in English nor its English-speaking country's law equivalent cited. Therefore, in order to fully understand all terms used in this thesis, a glossary with Czech equivalents of some terms is included in the appendix.

1.2 Small businessman

The main concern of this thesis is to create a software solution for small businessmen.

According to the Czech Ministry of Industry and Trade, a small businessman is a person who employs less than 10 employees and their assets or income do not exceed the Czech crown equivalent of 2 million Euros (in the current exchange rate this equals to approximately 51 million CZK). [1]

1.3 Tax records

Nearly every small businessman qualifies, by definition, for keeping tax records, as long as they do not exceed a yearly income of 25 million CZK. [2]

The tax records' primary concern is to determine the income tax base. [3]

The rules for keeping tax records are described in the Income Tax Act, § 7b. The act generally sets rules for evaluating the income tax based on types of income, expenses or assets. The rules for the form of keeping tax records are not strictly given by law, as the Income Tax Act essentially only states that the small businessman should keep income and expense records, claim and debt records, and asset records. The only firmly set rule is to keep all records for the whole tax period. [4] There are, however, generally accepted guidelines for keeping tax records.

1.3.1 Income and expense records

The first and the main part of tax records is income and expense records. The income and expense records are usually kept in what is referred to in Czech as the “Income and Expense Log”. It is a simple table which should contain at least the following columns: [3]

- date
- invoice number
- name or description
- type (income or expense - usually two columns and the amount is written in the corresponding type)
- amount
- influence on income tax (yes/no)

Both income and expenses can be either monetary or non-monetary. Additionally, a third type of income could be considered, and that is an income earned in exchange, or an income in a foreign currency. In such a case, the income records have to be kept separately until the end of the tax period. Then, those records should be exchanged at the uniform exchange rate set by the Ministry of Finance. [3]

Non-monetary expenses are the expenses that reduce the taxpayer’s tax base, such as depreciation of fixed assets or creation of reserves. [3]

1.3.2 Asset records

Assets and property which can be used for more than one year, cost more than 80,000 CZK (as of 2020) and which a small businessman uses for business purposes, are called fixed assets. [5]

There are two types of fixed assets — tangible and intangible. Intangible assets can be added to the income and expense records as an expense. Generally, software, royalties, and research and development results are considered intangible assets. [3]

Tangible assets, however, have to be recorded completely separately, as they lose value over the time. Each asset is described in its own table, sometimes called a “card”. [3]

The bare minimum of data to be included in the asset card is as follows: [3]

- name or description
- initial value
- start date of use (usually the same as the date of purchase)
- method of depreciation

- depreciation group
- overview of depreciation amount for all elapsed tax periods,
- date and method of disposal

As mentioned above, each asset has to be put in a depreciation group. In total, there are six groups. They influence the depreciation period and therefore also the depreciation amount each tax period.

Depreciation group	Depreciation period (years)	Examples of assets
1	3	office equipment, computers, tools
2	5	cars and trucks, furniture, televisions
3	10	air conditioning, elevators, boats
4	20	wooden buildings, gas pipelines
5	30	roads, production buildings, bridges
6	50	department stores, office buildings

Table 1.1: Depreciation groups and their depreciation periods

The small businessman can choose between two methods of depreciation, even or expedited, based on which one seems to be the best for their situation. Each asset owned by a businessman can be depreciated using a different method, but once a method is chosen, it is impossible to change it. [3]

Even depreciation

To count the amount to be depreciated each year using the even depreciation method, the following equation is used [4]:

$$A = V \cdot \frac{R}{100}, \quad (1.1)$$

where A is the amount to be depreciated, R is the annual depreciation rate and V is the initial value of the asset.

The annual depreciation rate is a percentage value which depends on the depreciation group of the asset. In 2022, the values were as follows [4]:

Depreciation group	R_1 (%)	R_2 (%)	R_z (%)
1	20,0	40,0	33,3
2	11,0	22,25	20,0
3	5,5	10,5	10,0
4	2,15	5,15	5,0
5	1,4	3,4	3,4
6	1,02	2,02	2,0

Table 1.2: Annual depreciation rate based on depreciation group

where R_1 is the annual depreciation rate for the first tax period (year) and R_2 is the annual depreciation rate for the rest of the depreciation period. R_z is the annual depreciation rate for the increased initial price, which occurs when the initial price is increased based on a technical evaluation.

Expedited depreciation

To count the amount to be depreciated each year using the expedited depreciation method, the following equations are used [4]:

For the first tax period:

$$A = \frac{V}{K_1}, \quad (1.2)$$

where A is the amount to be depreciated, V is the initial value of the asset and K_1 is the depreciation coefficient for the first tax period (year).

From the second tax period on:

$$A = \frac{2 \cdot V_r}{K_2 - n}, \quad (1.3)$$

where A is the amount to be depreciated, V_r is the residual value and K_2 is the depreciation coefficient for the rest of the depreciation period.

The depreciation coefficient depends on the depreciation group of the asset and in 2022, the values were as follows [4]:

Depreciation group	K_1	K_2
1	3	4
2	5	6
3	10	11
4	20	21
5	30	31
6	50	51

Table 1.3: Depreciation coefficients based on depreciation group

In the case of increased initial price, the coefficients are the same as K_1 , the depreciation coefficients for the first tax period.

Disposal of assets

An asset can be disposed of because of three different reasons:

- the asset has been sold
- the asset has been disposed of (liquidated)
- the asset has been gifted

If the asset was sold or liquidated, the whole residual amount will be considered an expense (in other words, the tax base will be reduced by this residual amount). However, in the case of liquidation, the businessman should be able to provide a proof that the asset was liquidated. [3]

In the case of gifting the asset, its residual value cannot be claimed as an expense. [3]

Minor assets

Tangible assets with a value under 80,000 CZK [4] are recorded as an expense and are therefore not depreciated. The small businessman should, however, still keep a record about them, including the date when they stopped using, or liquidated, such asset. [3]

1.3.3 Claim and debt records

According to the Income Tax Act, the last records that the businessmen are obliged to keep, are claim and debt records. At the end of the tax period, there should be a way to know if there are any invoices issued either by or to the small businessman, that have not been paid yet — these records are referred to as claims and debt. [3]

Once these invoices are paid, they become income (in the case of claims) or an expense (in the case of debt), and they can be deleted from the claim and debt records. [3]

1.3.4 Income tax

The base for income tax is the amount by which the total income exceeds total expenses in the corresponding tax period. It concerns only the income and expenses with an influence on the income tax. Therefore, it is financially advantageous to maintain the difference between income and expenses as little as possible. The tax itself is then 15 % of this base. [4] Alternatively, if the difference between income and expenses is too big, the businessman can opt for using the “flat-rate expense” option, which means that the total expenses equal to a percentage of the total income. The percentage depends on the field of the business. [3]

At the end of a tax period, the sum of depreciated amounts of all assets is considered an expense to be added to the total amount of expenses. [3]

1.4 Flat tax

Small businessmen can opt for either paying a flat tax rate or keeping tax records. The flat tax brings some advantages, namely doing no tax returns at the end of the tax period. [6] However, to qualify for the flat tax one must meet several conditions. According to the Czech Income Tax Act, §7a, such a person:

- is not a taxpayer (has only tax-exempt and non-taxable income)

- their income from independent activities does not exceed 1 million CZK
- their income from capital assets, rental income and other income described in §10 does not exceed 15,000 CZK,
- is not a partner of a public company or a general partner of a limited partnership.

Because this tax is strictly set and does not depend on the businessman's revenue, many people who qualify for paying flat tax still rather opt for tax records keeping, especially people with a really low income. This solution is not financially beneficial for those people, as they pay much less in taxes when they keep tax records. [6]

1.5 Bookkeeping

Instead of keeping tax records or paying flat tax, any small businessman can opt for doing bookkeeping if they wish, even when they are not obliged to do so. Bookkeeping can be required from a small businessman in the case when the businessman's turnover in the previous year exceeded 25 million CZK, if they are required to do so by a special legal regulation, or if they are partners associated in the company, where at least one of the partners associated in this company has to do bookkeeping. [2]. In this case, the small businessman ordinarily contacts an accountant to do the bookkeeping for him.

Chapter 2

Analysis

As mentioned in the previous chapter, if a small businessman is a flat-rate taxpayer, their income tax is equal to the flat tax, therefore there is no need for a software for this option. Bookkeeping is rare among small businessmen, and since it would require many more functionalities in the software, it will not be covered by this application. Moreover, keeping the books is something that is made exclusively by accountants, for whom this application is not intended. Therefore, its aim will be keeping tax records.

2.1 Consultation with a field expert

As part of a thorough analysis, an expert in the field of accounting was contacted to be consulted about tax records and to answer some questions. This professional is an accountant, but he also keeps tax records for some of his clients. He agreed to do user testing as well, and is listed as Person A in the Testing chapter.

As a result of this consultation, it was decided that the application should be made only for non-taxpayers, as the records for taxpayers are much more complicated to keep. It was suggested that small assets should be kept in a separate table in addition to the Income and Expense Log, to make marking of disposal possible. Data gathered in the Theory chapter was reviewed and validated.

The expert uses the Účto tax record program, which he generously let us use for a moment for a deeper analysis and further understanding of it.

2.2 Competitive analysis

In the following section, some popular solutions for keeping tax records are listed, as well as the experience gained while using one of them.

2.2.1 Excel

It is completely legal to use Microsoft Excel or any other spreadsheet software to keep tax records, as the law does not specify how exactly should the records be kept. An indisputable advantage of this approach is that spreadsheet software is free and has a well known user interface. There are numerous templates available to download and everything is customizable.

The main disadvantage of this approach is that it is impossible to check the accuracy of the data. Moreover, it does not have an option of generating a tax return form. Lastly, it is crucial for the user to be familiar with the complex asset records rules and to be able to do any necessary changes manually.

2.2.2 Money S3

The Czech company called Solitea offers a great number of software solutions for accounting. [7] The Money S3 software has, in addition, a module for both income and expense records and asset records. [8]

Even though Money S3 seems like a good solution, some disadvantages can be found. Due to the fact that this software is primarily made for accountants, there are a lot of possibly confusing or unnecessarily detailed terms and options. The tax record package is included in the cheapest version of the program, but it is still not free. [9]

It can be argued that another disadvantage is the fact that this is not a web application and the data therefore cannot be accessed from any device without downloading the application.

2.2.3 Účto

Účto, like Money S3, covers the whole tax records keeping process, including asset records.

The program is very old, first created for personal use in the year 1990 and introduced to the public in the year 2000. [10] In the logo of this software, displayed on their official website, there is even a text proclaiming “Simple accounting účto”, [10] even though the term “simple accounting” is no longer used. It was replaced by the term “tax records” in the year 2003. The term simple accounting is, by law, still sometimes used by some types of organizations. [4]

Despite its very outdated design, the application is very popular and generally liked by users. It updates the rules and adds new functionalities every year based on changes in laws and new regulations. [11]

For some people, especially those who would like to try the software before committing to it, a big disadvantage could be that there is no free version of this software, only a demo. [12] The price is, however, quite low compared to payed versions of other software and applications. Each update is also monetized.

As mentioned above, this application was briefly tested as part of the analysis.

The fact that Účto is a DOS application [13] brings both advantages and disadvantages. The main advantage is that the application is extremely fast. There is never any waiting time, even when loading records dating back to more than twenty years (as was shown by the field expert). Undoubtedly, the biggest disadvantage is its look — the application still looks more or less the same as it did more than 20 years ago, which can discourage some, especially young people, from using it. In addition, it is clear that user experience was not a priority, as some items are hidden under many layers of nested subitems of the menu.

On the other hand, it offers explanation of the majority of terms, allows the user to keep many types of records used less often (such as wages) and keeps the data in well-arranged lists.

2.2.4 iDoklad

iDoklad is a new, modern and simple solution to keeping tax records. Its interface is, unlike some other solutions mentioned, very user-friendly and intuitive. Its main purpose is, however, not tax records. This online web application was designed for businessmen and firms to create, receive and store invoices. The invoices are then stored in cloud, so the user can access the data from any device.

Income and expense records is a natural progression of these functions, as the businessmen's invoices are all stored in the application after all. The two values are also visualised in clear and useful graphs throughout the application.

It is, however, not possible to keep asset records in iDoklad. The application also contains a number of functions that are not essential for keeping tax records. [14]

2.2.5 Fakturoid

Fakturoid is a web application that works on a similar principle as iDoklad. It is much more simplified and offers less functionalities than iDoklad, however, this does not affect the functionalities for tax records keeping. The user can send invoices and add expenses, the data is then displayed in the form of a table and in the form of a graph. Identically to iDoklad, asset records cannot be kept in Fakturoid. [15]

2.2.6 Summary

On the basis of this competition analysis, some clear objectives for the tax record application are set.

The current solutions are either very old, not intuitive for new users and unnecessarily complicated, expensive or do not cover all main parts of the tax records keeping process. The application that will be the result of this thesis therefore should be intuitive even for people with no background in accounting, simple (without any unnecessary and confusing options). At the same time, it should cover every major part of the tax record keeping process, including asset records.

2.3 Requirements

2.3.1 Functional requirements

In the application, it should be possible to:

- FR1: keep income and expense records
- FR2: keep asset records
- FR3: keep claim and debt records
- FR4: automatically depreciate fixed assets
- FR5: filter and sort records by different criteria
- FR6: display all necessary information for filling in tax return form

2.3.2 Non-functional requirements

The application should:

- NR1: run on Windows 10 and 11
- NR2: be intuitive and easy to use even for users who are new to tax records
- NR3: have a modern look
- NR4: run quickly even with years worth of records
- NR5: have a solution for updating rules in case of changes in the law
- NR6: be available to users to download

2.4 Actors

This application will only have two actors — user and system. By “user” is understood any person using the application, “system” is the logic behind it.

2.5 Use cases

The following section will list all of the use cases created for this application. At the end of this section, use case diagram can be found.

2.5.1 UC1: Adding a new item for income and expense records

Actors: user, system

Preconditions: user is on the income and expense list page

Postconditions: a new income or expense item is created

Flow:

1. User clicks on the “New item” button.
2. System opens a new window with a form.
3. User fills in the “Income or expense item” form.
4. User clicks on the “Save” button.
5. System checks if all required data is filled in.
6. System adds a new item to the income or expense list.
7. System recalculates total income and expenses.
8. System closes the window with the form.

2.5.2 UC2: Adding a new card for asset records

Actors: user, system

Preconditions: user is on the asset list page

Postconditions: a new income or expense item is created

Flow:

1. User clicks on the “New item” button.
2. System opens a new window with a form.
3. User fills in the “Asset card” form.
4. User clicks on the “Save” button.
5. System checks if all required data is filled.
6. System adds a new item to the asset list.
7. System closes the window with the form.

2.5.3 UC3: Automatically depreciating assets and adding it to expenses

Actors: system

Preconditions: a year passed since a start date of use of an asset

Postconditions: appropriate depreciation amount is added to expenses as a new item

Flow:

1. System calculates the asset amount to be depreciated according to depreciation group, method of depreciation and residual value.
2. System creates a new expense item.
3. System adds a new item to the income or expense list.
4. System recalculates the total income and expenses.

2.5.4 UC4: Adding a new item for claim and debt records

Actors: user, system

Preconditions: user is on the claim or debt list page

Postconditions: a new claim or debt item is created

Flow:

1. User clicks on the “New item” button.
2. System opens a new window with a form.
3. User fills in the “Claim or debt item” form.
4. User clicks on the “Save” button.
5. System checks if all required data is filled in.
6. System adds a new item to the claim and debt list.
7. System closes the window with the form.

2.5.5 UC5: Accessing income and expense list

Actors: user, system

Preconditions: user is in the application

Postconditions: user can see all income and expenses in the form of a table

Flow:

1. User clicks on the “Income and Expense Log” button.
2. System opens a new window with a list of all the recorded income and expenses.

2.5.6 UC6: Accessing asset list

Actors: user, system

Preconditions: user is in the application

Postconditions: user can see all of the income and expenses in the form of a table

Flow:

1. User clicks on the “Asset records” button.
2. System opens a new window with a list of all recorded assets.

2.5.7 UC7: Accessing asset card detail

Actors: user, system

Preconditions: user is on the asset list page

Postconditions: user sees detailed information about an asset

Flow:

1. User clicks on a row in the asset list.
2. Systems expands the row to show the card detail.

2.5.8 UC8: Accessing claim and debt list

Actors: user, system

Preconditions: user is in the application

Postconditions: user can see all of the claim and debt in the form of a table

Flow:

1. User clicks on the “Claim and debt records” button.
2. System opens a new window with a list of all recorded claims and debt.

2.5.9 UC9: Marking a payment of a claim or a debt

Actors: user, system

Preconditions: user is on the claim and debt list page

Postconditions: the selected claim or debt is deleted, a new income or expense item is created

Flow:

1. User selects a row in either of the lists by clicking on it.
2. System enables the “Edit” button.
3. User clicks on the button.
4. System asks user to fill in the date of payment.
5. System creates a new income or expense item.
6. System deletes the record marked as payed.

2.5.10 UC10: Editing records

Actors: user, system

Preconditions: at least one expense, income or asset is added

Postconditions: data in corresponding record is changed

Flow:

1. User selects a row in either of the lists by clicking on it.
2. System enables the “Edit” button.
3. User clicks on the button.
4. System opens a new window with a form filled with data.
5. User makes the desired changes.
6. User clicks on the “Save” button.
7. System checks if all of the required data is filled.
8. System updates corresponding record.
9. System closes the window with the form.

Alternative flow:

5. (a) User tries to change the amount of a partially or fully depreciated asset
 1. System does not allow changes in price.
 2. System does not allow changes in depreciation method and group
6. (a) User changed the amount of an income or expense.
 1. System recalculates the total income and expenses.

2.5.11 UC11: Deleting records

Actors: user, system

Preconditions: at least one expense, income or asset is added

Postconditions: corresponding record is deleted

Flow:

1. User selects a row in either of the lists by clicking on it.
2. System enables the “Delete” button.
3. User clicks on the button .
4. System asks user whether they are sure.
5. System deletes the item.

Alternative flow:

3. (a) User tries to delete a partially or fully depreciated asset
 1. System asks what is the reason for deleting an item (sale/disposal/gift).
 2. Based on the answer, system makes an appropriate depreciation calculation.

2.5.12 UC12: Filtering records

Actors: user, system

Preconditions: user opened one of the lists

Postconditions: list displays only data relevant to selected filters

Flow:

1. User chooses a filter or multiple filters.
2. Systems finds all of the items that are relevant to selected filters.
3. System reloads the table and only displays items that are relevant to selected filters.

2.5.13 UC13: Sorting records

Actors: user, system

Preconditions: user opened either of the lists

Postconditions: list is displaying data in an order that the user chooses

Flow:

1. User clicks on one of the arrows next to the name of a column.
2. System sorts the table according to the values in the appropriate column, either ascending or descending depending on which type of arrow (up, down) has the user chosen.

2.5.14 UC14: Displaying income and expense overview

Actors: user, system

Preconditions: at least one asset is added

Postconditions: system displays a brief overview

Flow:

1. User goes to the homepage.
2. System displays brief information about the total amount of income and expenses for the corresponding tax period.

2.5.15 UC15: Displaying necessary data for filling in a tax return form

Actors: system

Preconditions: user is on the homepage

Postconditions: user is on the data page

Flow:

1. User clicks on the “See more” button under the brief overview.
2. System opens a new page with a more detailed overview and all necessary information for filling in a tax return form.

2.5.16 Use case diagram

The following diagram describes the relationships between the user, the use cases and the application.

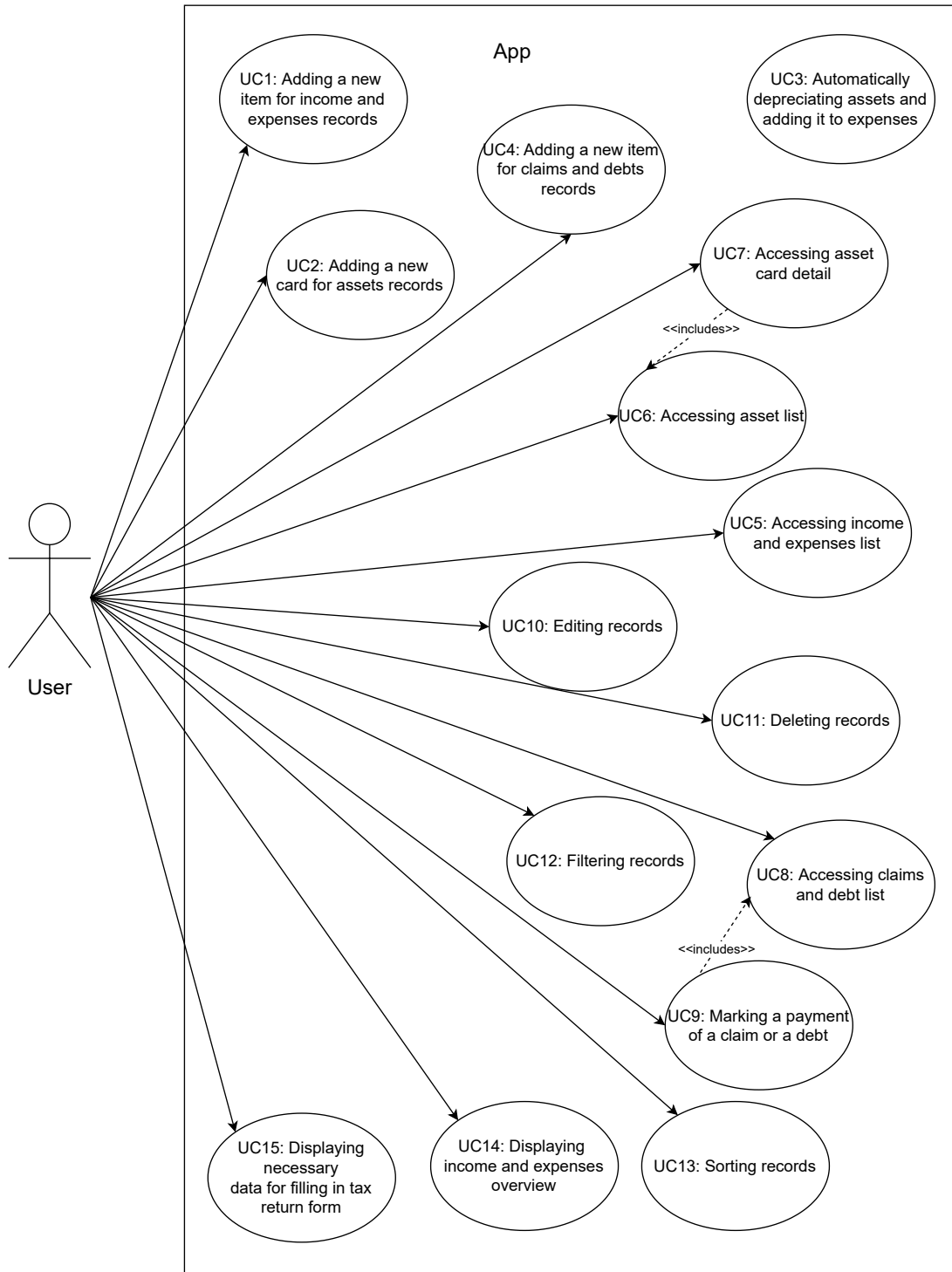


Figure 2.1: Use case diagram

In order for the diagram to be readable, several relations are not shown. Despite not being a part of the diagram, UC4, UC5 and UC6 each include use cases UC10, UC11, UC12 and UC13. In other words, functions for deleting, editing, sorting and filtering records will all be done at all of the “list” pages.

2.6 Requirement coverage

The following table shows which functional requirement is covered by which use case. It is evident that each requirement is covered by one or more use cases.

	FR1	FR2	FR3	FR4	FR5	FR6
UC1	X					
UC2		X				
UC3				X		
UC4			X			
UC5	X					
UC6		X				
UC7		X				
UC8			X			
UC9			X			
UC10	X	X			X	
UC11	X	X			X	
UC12					X	
UC13					X	
UC14						X
UC15						X

Table 2.1: Requirement coverage

Chapter 3

Technologies and design

The design of this application is based on the technologies used. Therefore, the technologies will be introduced first.

3.1 Technologies

The result of this thesis will be a desktop Windows application. It will be created using Windows App SDK with the framework WinUI 3. It will be written in C#, and the interface will be created using the XAML declarative language. For the database, SQLite was chosen.

All of the technologies mentioned are described below in detail.

3.1.1 Windows App SDK

Windows App SDK (Windows App Software Developer Kit), which used to be called project Reunion, is a set of APIs and tools that can be used for creating Windows applications for Windows 10 (version 1809 and higher) and Windows 11. [16] Such applications are consistent without the need of writing a version adaptive code. [17]

One of the features of Windows App SDK is WinUI 3. [17]

WinUI 3

Windows UI library, commonly referred to as WinUI, is a framework that can be used for Microsoft applications. WinUI 3, the newest version, was released at the end of the year 2021, which makes it a brand new solution for creating user interface. [18]

WinUI 3 contains modern components, which are designed to be consistent with the components of Windows 11 user interface. This provides a great advantage for the Tax records application, since some of the users will work with elements that they come in contact with on a daily basis, and therefore will be familiar with their behaviour. [18]

As WinUI 3 is a brand new framework, there is no design view available [19] at the moment. The application will therefore be developed using pure XAML. Fortunately, Microsoft created a useful desktop application called WinUI 3 Controls Gallery that contains XAML code snippets of different components. [20]

3.1.2 C#

C# is an object-oriented programming language developed by Microsoft, commonly used when developing Windows applications. [21]

3.1.3 SQLite

SQLite is an open source database solution. [22] It does not require database management system to be downloaded by the end user of the application, as it does not need a server to run. Instead, the database is essentially a part of the application the end user downloads. [23] In addition, SQLite is a free solution (unlike many other popular database engines), and it is in the public domain licence.[22]

On the other hand, SQLite has its limitations. It does not support a large amount of data [24] or multiple user environment.[25] However, as the application for keeping Tax Records will be a single-user application and will operate with relatively simple data, this has not been found a big disadvantage.

3.1.4 Entity framework

Entity Framework is an open-source object-relational mapper framework. It enables working with the database using .NET objects, thus eliminating any need for data-access code. In practice, this means that the developer can work with the database using only classes and their methods. [26]

Entity framework supports SQLite. [27] In the application, it will be handled by the package Microsoft.EntityFrameworkCore.Sqlite.

3.1.5 NuGet Packages

NuGet is a package manager. It allows the developer to manage all free and open-source extensions and frameworks. [28]

Packages that were necessary to use for this application are as follows:

- CommunityToolkit.WinUI.UI.Controls.DataGrid and Microsoft.Toolkit.Mvvm
 - Needed to use the datagrid component from Windows Community Toolkit.
- Microsoft.EntityFrameworkCore.Sqlite
 - Needed to use Entity Framework as a way to communicate with the database.

- Microsoft.Windows.SDK.BuildTools and Microsoft.WindowsAppSDK
 - Windows App SDK packages included in the blank app solution by default. The WinUI 3 framework is included in those packages.

3.1.6 Single-project MSIX Packaging Tools

In order to submit the application to the Microsoft Store, it needs to be packaged into a MSIX file. For doing this, the application “Single-project MSIX Packaging Tools for VS 2022” is used. [29]

3.2 Architecture

The application will be divided into three layers. The three layers are graphical user interface, business logic (with database connection and application logic component), and database layer.

3.2.1 Graphical User Interface

As mentioned above, the user interface will be created using the WinUI 3 framework. There will be one exception, which will be explained later in this section.

Homepage

The homepage will only contain a brief overview displaying total income and total expenses for the particular tax period. It will also include buttons leading to a page with an overview.

The main function of the homepage will be to navigate the user to the page. The user can also opt for using the navigation to get to the page instead.

Navigation

In order to make the application as user-friendly and clear as possible, it was necessary to provide the user with an option to get anywhere in the application from any page in one or two clicks maximum. To achieve this, it was decided to use the NavigationView component. The user will see all pages listed in a menu at the left of the page. This menu will be collapsible.

Lists

In the application, there will be four separate tables, one displaying all uploaded income and expenses, one with all assets, one with claims and debt, and one for only minor assets. Each will have their own page. The table will be created using

the DataGrid component from the Windows Community Toolkit. WinUI 3 does not have a table component, and in its Control Gallery, the user is advised to use the DataGrid component.

DataGrid supports some essential functions. It can be connected to a database, items can be filtered and each column can be easily sorted using arrows next to the column name.

In addition, by clicking on a row, the user can access “Row Detail”. This functionality can therefore be used in the asset table to access the asset card detail.

At the top of the table, there will be buttons — Filter, New item, Edit and Delete. There will also be a search field.

Buttons for editing and deleting will be disabled unless the user clicks on a row of the table.

The “New item” button will redirect the user to the desired form page.

Some pages will have special buttons. The claim and debt page will, in addition, contain the button “Mark payment”, which will add an end date for selected claim or debt, and create income or expense with the same data. For Minor asset, the user will be able to “Mark liquidation”. Both of these buttons will work similarly to edit and delete — they will be disabled until the user clicks on a row.

The minor asset list will only display expenses marked as such. They do not necessarily need their own table and could be listed by filtering the income and expense list. Putting them in a separate table, however, will make it easier to mark the date when it stopped being used, and to display only relevant data. The small assets can also be found in the income and expense table, as they are expenses nonetheless.

Forms

The three form pages will contain TextBoxes, ToggleSwitches and CalendarDatePickers customized for each form — income or expense, claim or debt and asset forms, covering all necessary data mentioned in the first chapter. The user can access each form from the corresponding table page by clicking on the icon “New item”, or using the navigation.

Overview page

A crucial part of the tax records keeping is to be able to generate an overview, which is able to display all records kept in a tax period and the total amount of income and expenses. The overview can also divide the calculations to cash and account transactions.

This information will be accessible in the overview page. The data will be recalculated after any income and expense record will be added. Moreover, the user will be able to choose the time period for which they want to display the data.

Hint page

The last page of the application to be mentioned is a hint page. There, the user will have a chance to access the user manual of the application.

In addition, it will also offer an overview about what some terms in the application mean to avoid any possible confusion.

All information will be static, written in plain text.

3.2.2 Application logic

The application logic component will include classes handling functions where more than one table from the database is needed. The main functionality in this version will be the automatic depreciation of assets.

Each time the user opens the application, it will check if a new tax period started. If it discovers that it did, the system will calculate the depreciation amount of all relevant assets based on the asset's depreciation group, method and the number of depreciation. All of the amounts will be then added as an expense. The user will get a simple notification when such an action occurs, in order to be informed and to be able to check the data and make some edits in the record if desired.

In particular, the check will have the following flow:

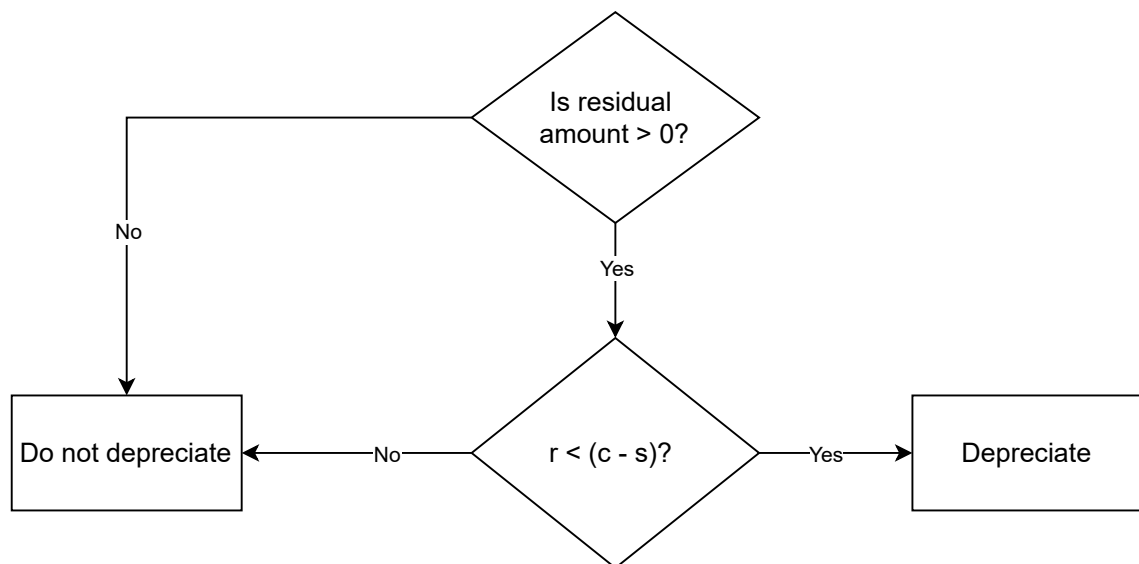


Figure 3.1: Depreciation flow, where r is the rank, c is the current year and s is the year when the user started using the asset

As was mentioned in the first chapter, the process of depreciating assets happens at the end of each year. Therefore, the difference between the current year and the year when the user started using the asset (in other words, how many years the asset has

been kept) should be the same number its depreciation with the highest rank (the number of times depreciation happened).

Using this logic, the application will select all assets that still have some residual amount left. Then, it calculates how long has the asset been kept, and compares it to the highest rank of its depreciation. If the rank is lower, it starts the depreciating process, until the numbers are the same.

This functionality will solve the automatic depreciation problem. It happens every time the user opens the application. It also means that the depreciation will not happen in the exact moment of the start of the tax period, but the first time the user will open it after that. This, however, should not be a problem. An average user will presumably open the application at least monthly, so the situation where a whole tax period, and therefore one instance of depreciation could be missed should not occur.

In addition, this component will add records about depreciation methods and groups into the database, in case they were not added already.

3.2.3 Database connection

To manage the connection with the database, SQLite Entity Framework Core Database Provider will be used.

The Database connection component will communicate not only with the database and the application logic component, but also directly with the interface. For example, data from the database can be displayed in the tables with the items/expenses and assets listed without much further processing. The same is true for the input from the forms. Updating (or editing) and deleting are also basic SQL functions that can work directly in the database connection component.

This component will consist of a general class covering the basic CRUD (create, read, update, delete) operations on the database. Then, each table in the database will have its own class that will inherit data from the general class with some more specific functions added.

3.2.4 Database

In order to efficiently edit any changes in the law regarding depreciation of assets, all information about depreciation groups and methods will be stored in the database, in addition to data about income, expenses and assets.

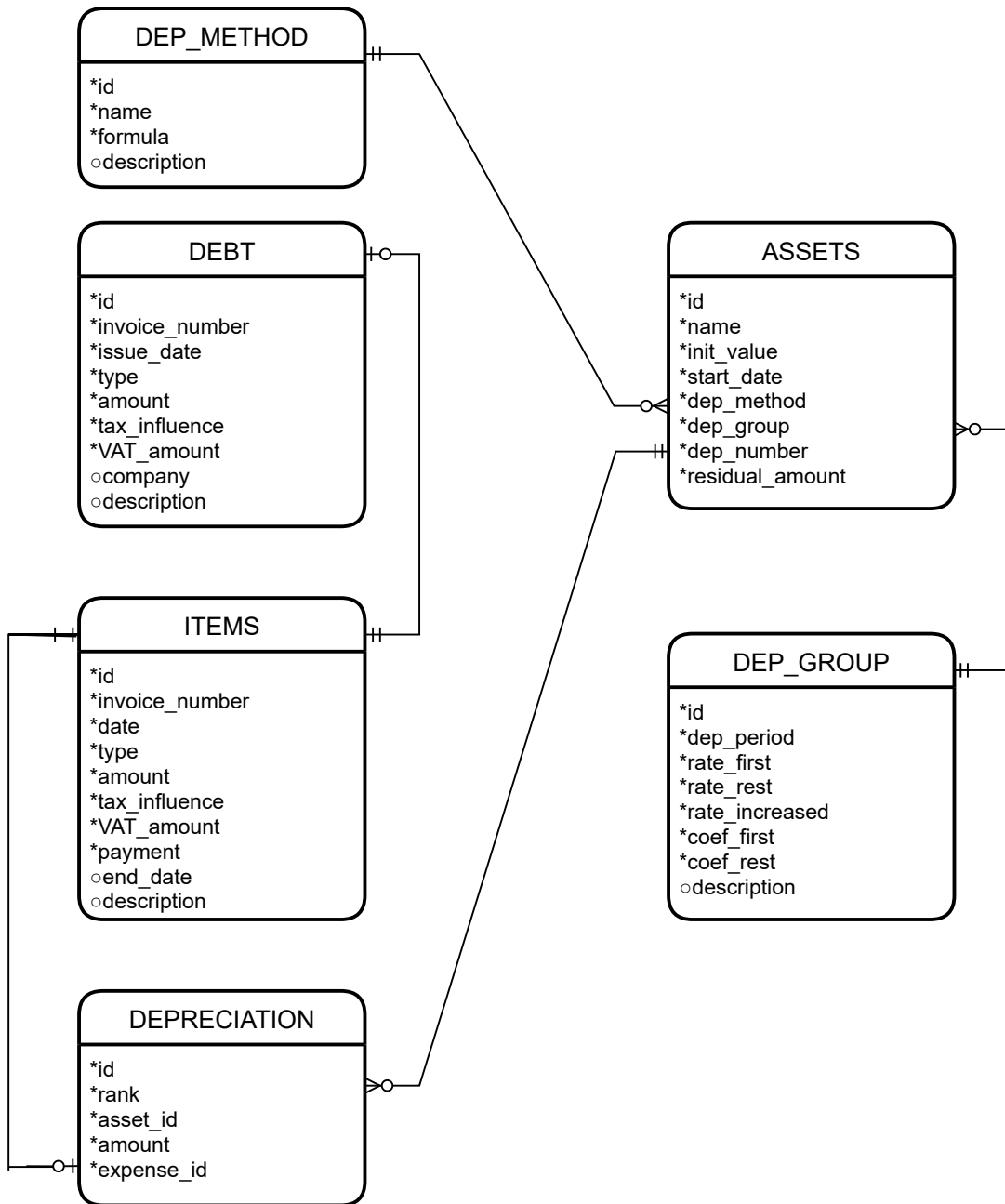


Figure 3.2: Database schema

The ITEMS table will keep data about all added income and expense items, the ASSETS table about all assets, and DEBT table about claims and debt. Information about depreciation groups — corresponding coefficients, rates and depreciation period, will be stored in the DEP_GROUP table. DEP_METHOD is a table with information about the two methods and formulas used for calculating depreciation. The last table is the table DEPRECIATION storing information about each depreciation across all assets. For each asset, it will also keep information about the sequence number of the depreciation, which is important for determining the rates and coefficients. Additionally, it will store an ID of the asset it is deducted from, and an ID of an expense item the depreciation ultimately turned into.

3.2.5 Architecture diagram

The following diagram visualizes the architecture structure indicated above.

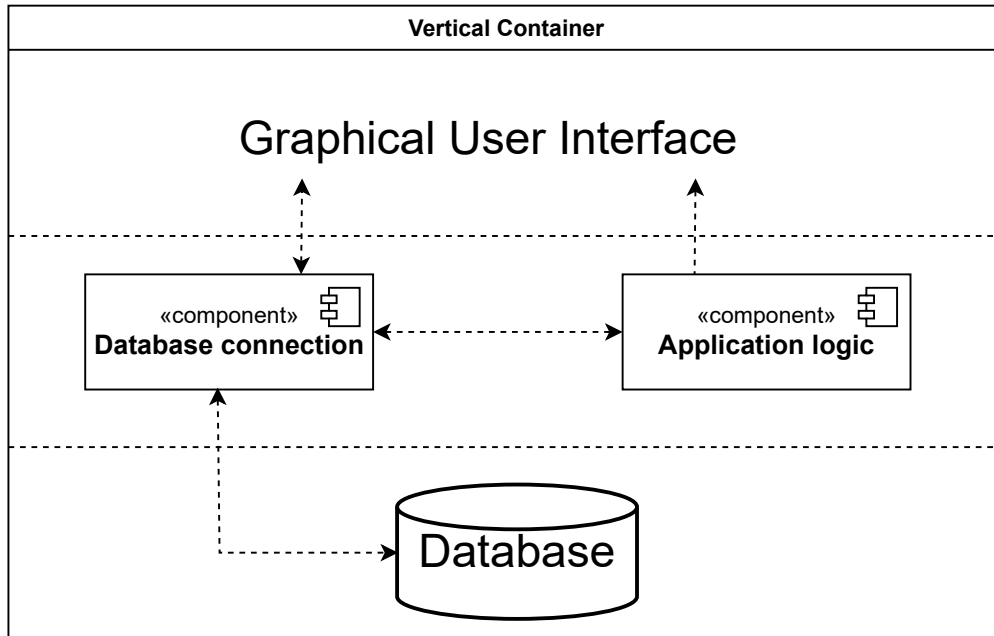


Figure 3.3: Architecture diagram

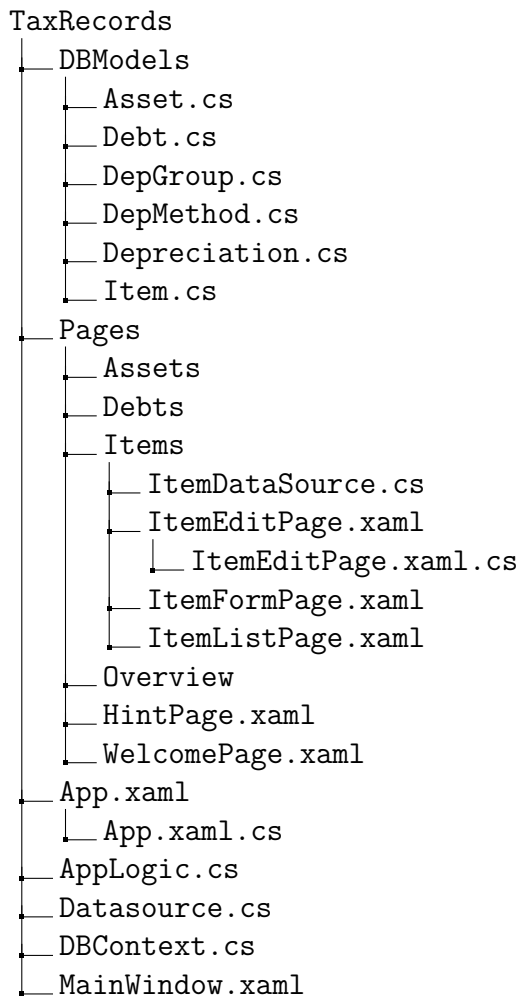
The Database connection component will communicate with the database, either reading data or updating it. It needs to be connected with the interface (GUI), because the interface will display data from the database and the database connection component will take data from forms to send to the database.

The application logic component will communicate with the database connection component, in order to access data about assets, the correct coefficient or rate corresponding to its depreciation group and the depreciation method. It will also create new expenses. It is connected to the GUI component as well, because it displays the notification.

Chapter 4

Realization

Based on the proposed design, the application was implemented. This chapter will go into detail about its classes and methods they use, and mention some difficulties faced while programming. The finished program has the following structure:



Some directories are not pictured in full. The directory tree with all files pictured can be found in the appendix.

The diagram is simplified in order to minimize the number of arrows, which could make the figure unreadable. If within a logical group all classes have the same relationship, the arrow is shown pointing from or to the whole group, not each class.

4.2 Classes and methods

Because the class diagram would be too extensive if methods were added, the most important methods will be mentioned in this chapter.

4.2.1 Main window

The main window page represents the application window and includes elements that are visible all the time, regardless of what page the user visits. In this case, it means navigation. The class corresponding to the main window introduces a method for redirecting between different pages.

4.2.2 Classes representing pages

As shown in Figure 4.1, the “Pages” folder includes all pages of the application. They can be divided into several logical groups based on their functionality, and some standalone special cases.

As is standard in any modern C# application, each page consists of a XAML file, and a class file. [30] In the folder, there are also *DataSource* files for each list page, which are described in a different section below.

List Page classes

By List page is understood any page that uses the previously described datagrid component to display appropriate data from the database. In particular, the pages are as follows: *AssetListPage*, *ItemListPage*, *DebtListPage* and *SmallAssetListPage*. A special case of this type of page is the *OverviewPage*.

In each list page class, most methods are event handlers. In particular, the classes all include

- a method for loading data from the DataSource of the page,
- methods that according to which filter is selected, assign a filter rule from the DataSource of the particular page,
- a method for sorting the table based on the rules from the DataSource,
- a method from deleting a row from the corresponding table in the database,
- a selection changed method, which finds out which row of the datagrid table is selected,

- and a method redirecting to an edit page when clicking on the edit button.

AssetListPage has the deleting method extended by the option to liquidate the asset (based on the rules described in the first chapter).

Overview page class

The overview page is a special case of a list page. Instead of buttons for adding a new item, deleting, editing and searching, there are only two calendarDatePickers marked “From” and “To”. The user can select the time period for the data to be displayed. When first visiting the page, the dates are set from the beginning to the end of the current year.

First method loads all items (income and expenses) in the selected time period with tax influence that are not depreciations.

After that, another method first puts a sum of all depreciations in the selected time period on a new row under the listed items, and then adds a row with sums of all income and expenses in the table.

Form Page classes

Another group of pages is form pages — *AssetFormPage*, *ItemFormPage* and *DebtFormPage*.

In each form page class, there is a *validateInputs* method. It checks if all mandatory fields are filled in, and if the amount is a number value. If a violation is detected, it prohibits sending the data, and displays an error message in an infobox above the form.

Another important method, *sendToDatabase*, links each column in a correct database table to a value filled in the form, and creates new records with such data.

AssetFormPage has, in addition, methods for loading data from DepGroup and DepMethod tables, in order to put values in comboboxes for choosing a group and a method.

Edit Page classes

Edit pages are nearly identical to Form pages as regards the XAML definition. They, however, use slightly modified and completely new methods.

There is a method for loading all data from the database and making the fields filled in. The edit page gets information from the list page about which particular row was selected. Then, it asks the database for the data in the selected row, and links it to the fields in the form.

This type of class also contains the *validateInputs* method, as well as a modified *sendToDatabase* method. It, in this case, instead of creating a new record, only updates the row that is being edited.

The way how to navigate to edit pages is also different. Instead of moving the user to some page on the navigation, it opens a new page over the list page. In fact, those

pages are not included in the navigation at all, and they can only be accessed by clicking the Edit button on one of the list pages.

Homepage class

The homepage, called *WelcomePage* in the code, displays the sum of the income and expenses in the current year, and therefore contains methods that make it possible. It also includes a method that shows a popup window when a depreciation happens.

Hint Page class

The hint page class does not include any methods. As mentioned in the previous chapter, the whole page is created using the XAML language, utilizing Pivot and RichTextBlock components in particular.

4.2.3 Communication with database

Another logical group of classes enables communication with the database. They can be divided into two groups:

Data sources

For doing operations on the database such as querying, inserting and other communication, *DataSource* files were defined.

There is an abstract *DataSource* class defining operations that all other DataSource classes can or have to implement. It contains two abstract methods that are mandatory for other DataSources. The first method is *GetDataAsync*. This method determines from which table in the database to load the data and which relations to include. By default, this method needs to be implemented in order to specify which data should be displayed or further worked with.

The other abstract method is *DBToTableModel*, an auxiliary method which enables data to be formatted before they are displayed in the datagrid. It converts an object of *DBModel* to an object of a class *DataSourceRow*. *DataSourceRow* is an auxiliary class defined for each DataSource. It would not necessarily have to be mandatory, but it was determined that every current and possibly future table would have to include at least one date or amount column, and those are precisely the fields that need to be formatted.

Additionally, the DataSource class defines virtual methods which can be extended or overwritten by other DataSources if their list pages call for such a functionality. The virtual methods are sorting, filtering, searching and loadData.

The other DataSources are each linked to a list page. In particular, they are *AssetDataSource*, *ItemDataSource*, *DebtDataSource*, *SmallAssetDataSource* and *OverviewDataSource*. They implement the abstract methods and any necessary virtual methods.

Database model classes

Database models create the database tables and introduce them to the application.

There is a separate database model defined for each table of the database, which was described in the previous chapter. Each model includes a definition of all its rows in accordance with the entity framework standards. In addition, the depreciation method and group models include a constructor method. This constructor makes it easier to add data in those tables when the application is opened for the first time, thus making the code appear clearer.

All database models are then initialized in the *DBContext* class, as is customary in entity framework applications.

4.2.4 Application logic class

In the previous chapter, a rough idea was given what functionalities will be covered by the application logic component. In this section, only a brief recapitulation is given, as the implementation did not differ from the design.

The application logic component is covered by the *AppLogic* class. As mentioned previously, this class has two main roles — to fill the *DepGroup* and *DepMethod* tables with data, and to manage depreciation.

The table filling part is all covered by one method. It checks whether the tables are empty, and if they are, it adds all information using the constructors mentioned above.

The functionality of calculating and doing depreciations is divided into several methods — some handle the calculations, one assigns the calculation based on the rank and the depreciation method and group, one is for creating a new expense and one for creating a depreciation. The method *doAutomaticDepreciations* includes the logic described in the previous chapter, and utilizes the above methods to handle the process of depreciating assets.

Both methods are called during the initialization of the application in the *App* class.

4.2.5 App class

The *App* class includes a *onLaunched* method, which calls the methods from *AppLogic* and also activates the Main Window.

4.3 Publishing

The application was packaged into several MSIX files (for different types of processors) using the packaging tool described in the chapter three. Then, these files were submitted to the Microsoft Store using a developer account. The application was

approved by Microsoft and is available in the store under the name “Daňová evidence SETR”. The users can download the application directly from the Microsoft Store, and it does not require any additional installation.

4.4 Problems with WinUI 3 discovered

Since WinUI 3 is a brand new framework, there are some components that are either not supported yet, or unusable because of bugs.

One unusable component is, for instance, the menu (MenuBar) component. There were several arguments for using NavigationView instead of a typical menu, but the main one was the fact that even though a menu is a part of the WinUI 3 package, there is a bug that makes it impossible to use. [**chyba**] The bug caused the program to display an error when an flyout item or subitem was added.

An even bigger challenge was to create form validations. If any validations need to appear, they are listed in an InfoBox on the form or edit page. The reason for this is the fact there was no component to display validations under the incorrectly filled-in field.

Chapter 5

User acceptance testing

This application was developed to be user-friendly, for people who may not have any accounting background and even for those who do not use computer very often. Four people were asked to test the application. They all have a different background and set of skills and therefore should be able to evaluate the application from different standpoints.

5.1 Testers

Person A

Person A is an accountant, therefore they are very familiar with the rules of tax records. He is hesitant to use new technologies. He will hopefully be able to comment on accuracy of the application.

Person B

Person B studied economics in addition to working as a small businessman in the past. Even though he is not currently working in the field, he still knows more about tax records than an average person. At the same time, he has worked as an IT administrator for many years, having a good command of computers.

Person C

Person C works as a programmer. He knows nothing about the process of tax records. He should be able to evaluate the application from the technical standpoint.

Person D

Person D is neither skilled with computers, nor has any experience with accounting. The aim of the application is to be clear and intuitive, and so even such a user should not have any major problem during the testing process.

5.2 Process

The users were presented with an application pre-filled with a number of records. The application was set up in a way that the automatic depreciation will be triggered after entering the application. They also had the user manual available, as a real user would have it as well. The testing was conducted in the Czech language, as some subjects do not speak English. The following section contains an English translation of the instructions they were given.

5.2.1 Briefing

At the beginning of the test, the users were given the following briefing:

- Have you ever used a similar application? If you did, which one? Is there anything you did not like when using the application?
- I will read you a set of instructions. Follow them as best as you can.
- Before asking me for help with any task, try to see if you can find any help in the application itself.
- During the test, you may comment on anything that you find unclear or confusing.
- If you wish to explore the application further, you will be given an extra time at the end of the test.

5.2.2 Scenario

Depreciation

1. Open the application.
2. After seeing a message on the homepage, follow its instructions.

Adding items

1. You want to add a record about your new computer. The computer cost 20,000 CZK with tax, you bought it today and started using it immediately.
2. (If they try to add it as a fixed asset) Now, you should see a pop up window telling you that the price is too low to keep it as an asset. Follow its instructions and add the computer as an expense.
3. Apart from the computer, you also bought and started using a new car today. It cost 500,000 CZK with tax, and you decided that for you, it makes more sense to depreciate it expeditiously. Add such a record.
4. Because you bought a new car, you sold your old one. Add an income of 200,000 CZK. Such an income has an influence on tax.

Lists

1. You realize that the new car actually cost 600,000 CZK. Try to change this detail.
2. As mentioned earlier, you sold your old car. The old car was not fully depreciated yet. Find the old record and try to do necessary changes.
3. You need to find a particular record. You do not remember much detail about it, you just know that it is an income of 20,000 CZK from February or March of 2020. Try to find this record.
4. Last week, the company Otec and synové s.r.o. finally payed off the invoice no. f789 you issued to them last year. Find this claim and mark the payment.
5. You remember that last month, your phone got destroyed. You have recorded the phone as a minor asset in the past. Please mark that you no longer use the phone.

5.2.3 Debriefing

At the end of the test, the persons were asked the following additional questions:

- Do you find this application generally easy to work with?
- If you worked with any other similar application before, is this one more or less confusing?
- Can you imagine using this application for keeping tax records?
- Did you find the More information and Hint pages useful?
- Would you like to see more information on the Homepage?
- Is there any other functionality you would like to see in the application?

5.3 Results

5.3.1 Person A's comments

As mentioned in the Analysis chapter, person A uses Účto for some of his clients. He is used to having more accounts within one application, but this application was made for personal use only.

As for his remarks, he wanted the amounts in the bank and in cash to be added in order to see them immediately, so that the user would not have to count it manually. He also remarked that in the small asset table, there was no option to add a new small asset and the user had to go the Income and Expense Log (or to click New income or expense in the navigation).

He could not find the option to liquidate an asset - it did not occur to him that it could be hidden under the Delete button. He recommended renaming the button.

The last major remark is that by going to the hint page while creating a new item, for example to see which depreciation group to choose, all data in the form is deleted and the user has to start over. He suggested displaying the hint page in a popup window.

In Účto, he is used to using printing sets, that is, being able to add a tag to records and then printing those with the same tag. It was something that he missed in this application.

As person A was the first to do the testing two small bugs were discovered, which were fixed immediately afterwards.

Generally, his experience was positive. He especially praised the ease of navigation and good user experience, stating how quickly he got familiar with the application.

5.3.2 Person B's comments

Person B recalled using some application for keeping tax records approximately twenty years ago, but could not recall its name or whether he liked the experience.

Most of his remarks concerned the user interface. He mentioned that the font in the application was rather small and was not easy for him to read. Some components of the user interface were also, according to him, not distinguished enough from the background. He would like to see examples of what kind of assets belong to each depreciation group directly on the asset form page, not only in the hint page.

He also could not find the liquidate asset option without a help.

During the debriefing, he mentioned that there should be a way to back up the data. He commented on the loading time of the tables — he did not find it too long, but definitely noticed it. This, unfortunately, is caused by the technologies used and cannot be fixed.

He appreciated the application's layout, calling it "simple".

5.3.3 Person C's comments

Person C have heard of some accounting programs, but not about any tax record programs.

He recommended all spaces and offsets in the application to be unified. Additionally, he proposed the idea of using icons on all buttons and next to the headers to improve the user experience. He said that the selected filter should be highlighted, because in the current state, the user might forget which filter was selected, or that any was selected at all.

Because, presumably, most of the time users will visit the application, they will do so to add a new record, he mentioned that the "add" functionalities should be higher on the navigation, and a link to them should be added to the homepage.

If he could add any other functionality in the application, he would like to see some graphs and statistics. He emphasised the importance of revising and updating the rules each year.

5.3.4 Person D's comments

Similarly to person C, person D has never used or even seen any similar application before.

Having no economics experience, she found it hard to understand the difference between asset records and the rest of the records. She often tried to make an incorrect action without looking at the hint page first, despite knowing it was there. However, once she studied the hint page more thoroughly, she seemed to understand most terms and functionalities.

Interestingly, instead of using filter and search functionalities on the income and expense list to find a particular record, she used the overview page, where, using the two datepickers, she viewed only items in the time period the task asked for. When asked whether she would appreciate this functionality in the income and expense page as well, she said she did not find it necessary.

Just like person A, she remarked that opening the hint page when filling in a form removed data from the fields.

On the whole, she liked the application's intuitiveness and found that the biggest issue for her was her lack of knowledge of the terms. She admitted that if she was to keep tax records, she would make an effort to get familiar with the terms before using the application.

5.3.5 Summary

After the user acceptance testing, several functionalities were added. In particular, a new button was added on the item list page that, upon being clicked, displays the sum of money in the bank and in cash in a flyout window. Icons were added on most buttons and the spacing was improved. Finally, an image and buttons leading to form pages were added to the homepage.

Unfortunately, WinUI 3 does not contain any component that would allow the hint page to be displayed in a popup window. There is a flyout component, but it can only be used on buttons, and not on the navigation items. To eliminate at least some of the problems, a hint button with the flyout was added both to the asset form page and the asset edit page.

The database backup and the opportunity to print the data could be both solved by adding a way to export the data into a CSV file. Unfortunately, this problem was too extensive to be solved within the remaining time to finish this project.

In general, the user acceptance testing was a success. Most remarks concerned details and not the core functionalities themselves. Every user said that they would use the application, and most mentioned the ease of navigation.

5.4 User manual

As mentioned at the beginning of the chapter, users were given a user manual, available here in the appendix. It was written in Czech, as users who use this application will be primarily, if not only, Czech. Tax records is, after all, a process specific to the Czech Republic and its laws.

Conclusion

The main concern of this project was to create a desktop application for keeping tax records, in particular the income and expense records, debt and claim records, and fixed asset records. Another requirement was to make the application user-friendly and intuitive, while remaining free, in order to stand out from the competition.

The finished program fulfilled these expectations. During the user acceptance testing, the testers praised the ease of navigation, which was one of the main objectives set. If there was confusion, it stemmed from the unfamiliarity with the terms used, not from the application itself. This showed the importance of the the hint page. Furthermore, the correctness of the processes was validated by two people who understand them.

WinUI 3 made it possible to create an application with a modern look and well-known functionalities. It included most components needed to implement all requirements, except for the datagrid. There were some surprising details that improve the user experience without a need of any implementation, such as the fact that the application automatically reacts to the theme and the accent colour set in the computer. There were, however, unsolvable issues discovered during the development process. The testers had some remarks regarding the components themselves, but they could not be improved. Generally, WinUI 3 seems to be a good framework for Microsoft application development, but first, it is necessary to fix some bugs and add a wider range of components.

While the application works well in its current state, there is still a potential for further development. The most pressing functionality to be solved is the database backup. In addition, printing sets could be added, and printing of each table could be possible directly in the application. In order for graphs to be added, some external library would probably have to be used, but it is definitely a possibility.

As for functionalities not mentioned by the testers, some other, not so common forms of records could be kept, such as wage records. Additionally, users could be allowed to upload their invoices into the application as a pdf file, and have an option to view or download them. Adding a functionality to keep records for several users would make the application accessible for accountants. Lastly, it would be helpful to allow the user to print a filled-in tax return form.

The application is available for free. In the following years, it will be revised continuously and any changes in the law will be reflected.

Bibliography

1. MINISTERSTVO FINANCÍ ČESKÉ REPUBLIKY. *Vymezení pojmu drobný, malý a střední podnikatel a postup pro zařazování podnikatelů do jednotlivých kategorií* [online] [visited on 2021-12-17]. Available from: <https://www.mpo.cz/assets/dokumenty/27300/28024/310086/priloha017.pdf>.
2. *Zákon o účetnictví (úplné znění). Předpis č. 563/1991 Sb. ve znění platném od 1. 1. 2022.*
3. DUŠEK, Jiří. *Daňová evidence podnikatelů 2021*. Grada Publishing, 2021. ISBN 978-80-271-3106-8.
4. *Zákon o daních z příjmů (úplné znění). Předpis č. 586/1992 Sb. ve znění platném od 1. 1. 2022.*
5. *Vyhláška č. 500/2002 Sb. ve znění platném od 1. 1. 2018.*
6. KUBÁTOVÁ, Alena. *Jak nejlépe evidovat svoji podnikatelskou činnost?* [Online] [visited on 2022-06-14]. Available from: <https://portal.pohoda.cz/pro-podnikatele/jak-zacit-podnikat/ucetnictvi-danova-evidence-pausal/ucetnictvi-danova-evidence-pausalni-vydaje-ci-p/>.
7. SOLITEA, a.s. *Účetní software* [online] [visited on 2022-01-08]. Available from: <https://solitea.com/cs-cz/ucetni-software>.
8. SOLITEA, a.s. *Volnost podnikatelům, klid účetním* [online] [visited on 2022-01-08]. Available from: <https://money.cz/>.
9. SOLITEA, a.s. *Money S3 Lite* [online] [visited on 2022-01-08]. Available from: <https://money.cz/komplety/money-s3-lite/>.
10. TICHÝ & SPOL. *O nás* [online] [visited on 2022-01-08]. Available from: <https://www.ucto2000.cz/default.htm>.
11. BLÁHA, Jan. *Ceník, aktualizace, licence* [online] [visited on 2022-01-08]. Available from: <https://www.ucto-tichy.cz/podpora/casto-kladene-dotazy/cenik-licence-aktualizace/>.
12. BLÁHA, Jan. *Ceník programu Účto Tichý* [online] [visited on 2022-01-08]. Available from: <https://www.ucto-tichy.cz/cenik-ucto/>.
13. BLÁHA, Jan. *Účto a Windows* [online] [visited on 2022-01-08]. Available from: <https://www.ucto-tichy.cz/podpora/casto-kladene-dotazy/ucto-a-windows/>.
14. SOLITEA, a.s. *Zjednodušte si život jako 200 000 podnikatelů před vámi* [online] [visited on 2022-01-08]. Available from: <https://www.idoklad.cz/>.

15. FAKTUROID S.R.O. *Jednodušší a chytřejší podnikání pro vás* [online] [visited on 2022-01-08]. Available from: <https://www.fakturoid.cz/>.
16. MICROSOFT CORPORATION. *WindowsAppSDK* [online] [visited on 2022-06-14]. Available from: <https://github.com/microsoft/WindowsAppSDK>.
17. MICROSOFT CORPORATION. *Windows App SDK* [online] [visited on 2022-06-14]. Available from: <https://docs.microsoft.com/en-us/windows/apps/windows-app-sdk/>.
18. MICROSOFT CORPORATION. *Windows UI Library Overview* [online] [visited on 2022-06-14]. Available from: <https://github.com/microsoft/microsoft-ui-xaml/blob/main/docs/roadmap.md%5C~#winui-3>.
19. MICROSOFT CORPORATION. *WinUI 3.0 no XAML designer view?* [Online] [visited on 2022-02-13]. Available from: <https://github.com/microsoft/microsoft-ui-xaml/issues/3656>.
20. MICROSOFT CORPORATION. *Windows UI Library (WinUI) 3* [online] [visited on 2022-06-14]. Available from: <https://docs.microsoft.com/en-us/windows/apps/winui/winui3/>.
21. MICROSOFT CORPORATION. *A tour of the C# language* [online] [visited on 2022-03-15]. Available from: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
22. SQLITE. *SQLite Is Public Domain* [online] [visited on 2022-03-15]. Available from: <https://www.sqlite.org/copyright.html>.
23. SQLITE. *SQLite Is Serverless* [online] [visited on 2022-03-15]. Available from: <https://www.sqlite.org/serverless.html>.
24. OWENS, Michael. *The Definitive Guide to SQLite*. Apress Media LLC, 2006. ISBN 978-1-4302-0172-4.
25. MUNZERT, Simon. *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. John Wiley & Sons, Inc., 2014. ISBN 978-1-118-83481-7.
26. ENTITYFRAMEWORKTUTORIAL.NET. *What is Entity Framework?* [Online] [visited on 2022-06-14]. Available from: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>.
27. MICROSOFT CORPORATION. *Database Providers* [online] [visited on 2022-06-14]. Available from: <https://docs.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli>.
28. MICROSOFT CORPORATION. *An introduction to NuGet* [online] [visited on 2022-06-14]. Available from: <https://docs.microsoft.com/en-us/nuget/what-is-nuget>.
29. MICROSOFT CORPORATION. *Package your app using single-project MSIX* [online] [visited on 2022-06-29]. Available from: <https://docs.microsoft.com/en-us/windows/msix/package/packaging-uwp-apps>.
30. MICROSOFT CORPORATION. *Implement navigation between two pages* [online] [visited on 2022-02-13]. Available from: <https://docs.microsoft.com/en-us/windows/apps/design/basics/navigate-between-two-pages>.

Attachments

A Contents of the enclosed CD

src.....	source code of the application
— readme.txt.....	description of the contents
— assignment.pdf	assignment of the Bachelor's Degree Project
— projecttext.pdf.....	text of the Bachelor's Degree Project

B Glossary

Assets records

Evidence majetku

Bookkeeping

Účetnictví

Claim and debt records

Evidence pohledávek a dluhů (dříve závazků)

Depreciation

Odpis

Fixed assets

Dlouhodobý majetek

Flat tax

Paušální daň

Flat-rate expense

Paušální výdaj

Income and expense records

Evidence příjmů a výdajů

Intangible assets

Nehmotný majetek

Minor asset

Drobný majetek

Small businessman

Drobný podnikatel

Tangible assets

Hmotný majetek

Taxpayer

Plátce daně

Tax records

Daňová evidence

C Uživatelská příručka

V této uživatelské příručce najdete podrobný návod, jak zacházet s aplikací Daňová evidence SETR (Simple and easy tax records).

C.1 Podporovaná zařízení

Aplikaci je možné spustit pouze na zařízeních s operačním systémem Windows 10 a vyšším. Aplikace pro spuštění nevyžaduje internet.

C.2 Přidání příjmu nebo výdaje

Příjmy a výdaje se přidávají pomocí stejného formuláře. Formulář lze nalézt v postranní navigaci pod záložkou Nový -> Nový příjem nebo výdaj, nebo po kliknutí na tlačítko "Nová položka" v záložce Deník příjmů a výdajů.

Jméno, částka, datum uhrazení, číslo faktury, druh platby a typ jsou povinné položky. V případě potřeby najdete další podrobnost v záložce Náповěda.

C.3 Přidání pohledávky nebo dluhu

Pohledávky a dluhy se přidávají pomocí stejného formuláře. Formulář lze nalézt v postranní navigaci pod záložkou Nový -> Nová pohledávka nebo dluh, nebo po kliknutí na tlačítko "Nová položka" v záložce Pohledávky a dluhy.

Jméno, částka, datum vystavení, číslo faktury a typ jsou povinné položky. V případě potřeby najdete další podrobnost v záložce Náповěda.

C.4 Přidání majetku

Formulář pro přidání nového majetku lze nalézt v postranní navigaci pod záložkou Nový -> Nový majetek, nebo po kliknutí na tlačítko "Nový majetek" v záložce Karty majetku.

Jméno, částka, datum začátku používání, odpisová skupina a způsob odpisu jsou povinné údaje. V případě potřeby najdete další podrobnost v záložce Náповěda.

C.5 Práce s tabulkami

Sekce Deník příjmů a výdajů, Pohledávky a dluhy, Dlouhodobý majetek, Drobný majetek a Přehled obsahují tabulky, ve kterých jsou vypsány položky. Všechny tabulky umožňují následující akce:

Všechny sloupce v tabulce je možné libovolně rozšiřovat, a to tak, že najedete myší na oddělovací čáru mezi názvy sloupců, dokud se vám nezmění kurzor, kliknete a potáhnete doprava nebo doleva.

Sloupce lze také libovolně přesouvat, stačí kliknout na Název sloupce, podržet a přesunout sloupec tam, kam chcete.

Všechny sloupce lze také seřadit. Seřazení je možné provést kliknutím na název sloupce bez podržení. Jedno kliknutí seřadí sloupec od vzestupně, dvě kliknutí sestupně a třetím se seřazení zruší. Pokud je sloupec seřazený, uvidíte vedle názvy odpovídající šipku.

Je-li tabulka příliš široká, takže nevidíte všechny sloupce, stačí kurzorem myši sjet ke spodu stránky, kde se objeví horizontální scrollbar, a podle libosti s ním posouvat obsah.

C.6 Deník příjmů a výdajů

Příjmy a výdaje se zaznamenávají v deníku příjmů a výdajů. Najdete ho v záložce Evidence faktur -> Deník příjmů a výdajů. Pokud chcete zobrazit pouze data z právě probíhajícího účetního období, nebo z období minulého, můžete využít filtry (fungování filtrů a další funkce jsou popsány níže).

C.7 Pohledávky a dluhy

Pohledávky a dluhy (dříve závazky) se evidují ve vlastní tabulce. Postup je velmi podobný, jako při evidování příjmů a výdajů, poue místo "Nový příjem nebo výdaj" vyberete "Nová pohledávka nebo dluh". Tabulku najdete pod záložkou Evidence faktur -> Pohledávky a dluhy.

Zaplacení pohledávky nebo dluhu

Pro zaznamenání zaplacení pohledávky nebo dluhu kliknete na příslušný řádek v tabulce pohledávek a dluhů, což by mělo zaktivnit tlačítko "Zaznament platbu". Toto tlačítko stisknete. Uvidíte vyskakovací okno, kde se vás aplikace zeptá na datum, kdy byl doklad zaplacen. Datum vyplníte a kliknete na tlačítko "Potvrdit".

V tento okamžik se pohledávka nebo dluh propíše do tabulky příjmů a výdajů a z tabulky pohledávek a dluhů zmizí. Nemusíte už nic dalšího vyplňovat.

Další možnost je pohledávku nebo dluh z tabulky smazat, a následně ručně vyplnit formulář pro přidání nového příjmu nebo výdaje, který by zaplacené pohledávce nebo dluhu odpovídal.

C.8 Karty majetku

Karty majetku jsou uchovávány v podobné tabulce, jako deník příjmů a výdajů, v záložce Majetek -> Dlouhodobý majetek. Oproti jiným tabulkám je tu změna v chování, pokud kliknete na řádek s položkou. Po takové akci se řádek rozbalí a nabídne vám přehled odpisů proběhlých na majetku. Zbytek informací o majetku můžete najít v řádku tabulky.

C.9 Drobný majetek

Drobný majetek se eviduje jako výdaj. Při vytváření výdaje je potřeba zaškrtnout, že se jedná o drobný majetek (políčko Drobný majetek? s možnostmi Ano/Ne). Pokud ho zkusíte přidat jako dlouhodobý majetek, aplikace vás upozorní a řekne, že máte záznam přidat jako výdaj.

Položky drobného majetku najdete v Deníku příjmů a výdajů, ale pro lepší přehlednost a evidenci mají i vlastní tabulku v záložce Majetek -> Drobný majetek. Do tabulky nemusíte majetek sami přidávat, propíše se tam automaticky.

Pokud v tabulce nemůžete drobný majetek najít, je možné, že jste při vytváření výdaje nezaškrtnuli, že se jedná o drobný majetek. Vraťte se do Deníku příjmů a výdajů, a příslušný výdaj upravte. Upravování položek je popsáno níže.

Vyřazení drobného majetku

Pro zaznamenání vyřazení drobného majetku kliknete na příslušný řádek v tabulce drobného majetku, což by mělo zaktivnit tlačítko "Vyřadit". Toto tlačítko stisknete. Uvidíte vyskakovací okno, kde se vás aplikace zeptá na datum, kdy byl majetek vyřazen. Datum vyplníte a kliknete na tlačítko Potvrdit. V tabulce drobného majetku nyní uvidíte i datum vyřazení v odpovídajícím sloupci. Majetek v tabulce zůstává. Pozor, tuto akci nelze provést v Deníku příjmů a výdajů!

C.10 Mazání a editace záznamů

Mazání a editace záznamů jsou funkce, které lze nalézt na všech stránkách s tabulkami kromě přehledu. Funkce jsou zakázané, dokud nekliknete na řádek s konkrétní položkou, kterou chcete odstranit nebo editovat.

Mazání

Pro odstranění záznamu stačí na záznam kliknout a poté kliknout na tlačítko Odstranit. Objeví se upozornění upozornění s dotazem, zda jste si jistí.

Kvůli evidenci není bohužel možné majetek odstranit, pokud už na něm proběhl odpis. Majetek ale může být vyřazen ze tří důvodů – prodej, darování nebo likvidace. Při kliknutí na tlačítko odstranit v případě majetku s proběhlými odpisy je tedy možné vybrat jednu z těchto možností. Aplikace poté automaticky provede operace nezbytné pro jednotlivé případy.

Editace

Po vybrání požadovaného záznamu a kliknutí na tlačítko Upravit se otevře stejný formulář, jako při zakládání nového záznamu, s předvyplněnými údaji o vybrané položce. Údaje můžete přepsat nebo smazat (v případě nepovinného pole). Kliknutím na tlačítko Uložit budou změny přepsány, a vrátíte se zpět na odpovídající tabulku.

Není bohužel možné editovat cenu, odpisovou skupinu ani způsob odpisu pro majetek, kde už odpis proběhl.

C.11 Filtry a hledání

Pro lepší orientaci v tabulkách lze využít filtry a hledání. Filtry, hledání i řazení sloupců popsané výše mohou všechny fungovat zároveň.

Hledání

V tabulce můžete hledat pomocí pole s textem "Hledat...". Když do něj začnete psát, data se automaticky začnou vyhledávat, nemusíte pro vyhledávání klikat na zvláštní tlačítko. Pokud text ve vyhledávání smažete, vyhledávání se zruší. Vyhledávání je možné pro jakákoliv slova, data nebo čísla v tabulce.

Filtry

Pro rychlé vyfiltrování některých dat je možné použít předpřipravené filtry. Připravené filtry jsou následující:

- Deník příjmů a výdajů
 - Letošní rok - položky s datem zaplacení v letošním roce
 - Minulý rok - položky s datem zaplacení v minulém roce
 - Příjmy
 - Výdaje
- Pohledávky a dluhy
 - Letošní rok - položky s datem vystavení v letošním roce
 - Minulý rok - položky s datem vystavení v minulém roce
 - Pohledávky
 - Dluhy
- Dlouhodobý majetek
 - Letošní rok - položky s datem začátku používání v letošním roce
 - Minulý rok - položky s datem začátku využívání v minulém roce
 - Aktivní - majetek, u kterého stále probíhají, nebo zatím neproběhly odpisy
 - Plně odepsané - majetek, kde už všechny odpisy proběhly
- Drobný majetek
 - Aktivní - stále využívaný majetek
 - Vyřazené - majetek, kde bylo zaznamenáno datum vyřazení

C.12 Odpisy majetku

Odpisy majetku probíhají v aplikaci automaticky, manuální odepisování není v této verzi aplikace možné. K odpisu dojde při prvním spuštění aplikace od začátku nového účetního období. Když k odpisům dojde, zobrazí se upozornění ve formě vyskakovacího okna. Odpisy najdete v tabulce Deník příjmů a výdajů, kde budou zařazeny jako výdaje s automaticky vygenerovanými předvyplněnými údaji. Tyto údaje můžete poté podle libosti upravit pomocí funkce Upravit (odkaz). Nedoporučujeme měnit cenu a typ záznamu (odkaz?).

C.13 Stránka Přehled

Stránka přehled obsahuje tabulku s přehledem evidence. Zobrazuje pouze položky, které jsou daňové, protože takové položky jsou důležité pro vyplnění daňového přiznání. Oproti deníku příjmů a výdajů jsou částky pro lepší přehled odděleny do zvláštních sloupců podle toho, jestli je položka příjem, nebo výdaj. Nad tabulkou najdete prostor, kam můžete vyplnit, od kdy do kdy chcete data zobrazit. Na předposledním řádku jsou pak sečteny všechny odpisy majetku, pokud v daném období proběhly, a na posledním řádku najdete součet všech daňových příjmů a součet všech daňových výdajů.

C.14 Stránka Náповěda

Stránka Náповěda obsahuje přehled některých pojmů v aplikaci a jejich základní vysvětlení. Najdete v ní také tuto příručku.

D Účto

Účto 2022		19.06.2022	
145 PENEŽNÍ DENÍK B:393119.98 H:100288		CelkemZ Výkon P1	
Datum	Doklad>	Druh Text>	NázevFirmy>
25.03.22	b/3	DPP záloha na daň z příj	
25.03.22	b/3	OS osobní výběr	
25.03.22	V/64	PR nafta	
28.03.22	b/3	JVY leasing - minibagr	
29.03.22	f/22008, P/8	PS autodoprava, zemní p	
29.03.22	f/22005, b/3	PS autodoprava	
29.03.22	f/22007, b/3	PS autodoprava	
30.03.22	V/65	PR AdBlue	
31.03.22	b/3	PKM úroky	
31.03.22	V/66	M výplata DPP	
			-1500.00
			-1500.00
			-1200.00
			-6828.00
			23837.00
			4783.13
			1339.47
			-266.00
			3.67
			-10000.00

Figure 1: Income and expense log. It contains only data form the current tax period (year)

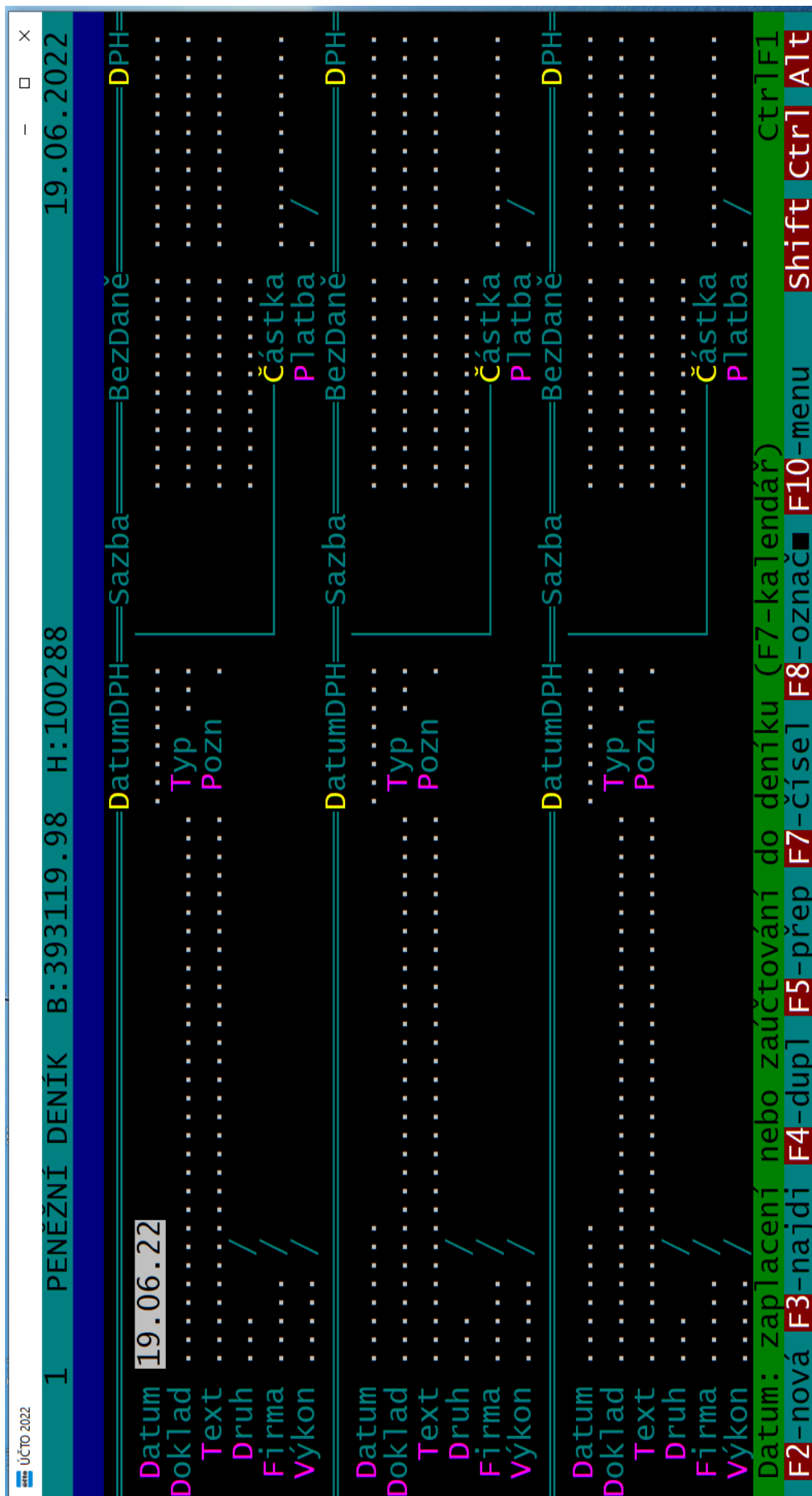


Figure 3: Adding new items. Several items can be added at a time, but saving them is done in another tab

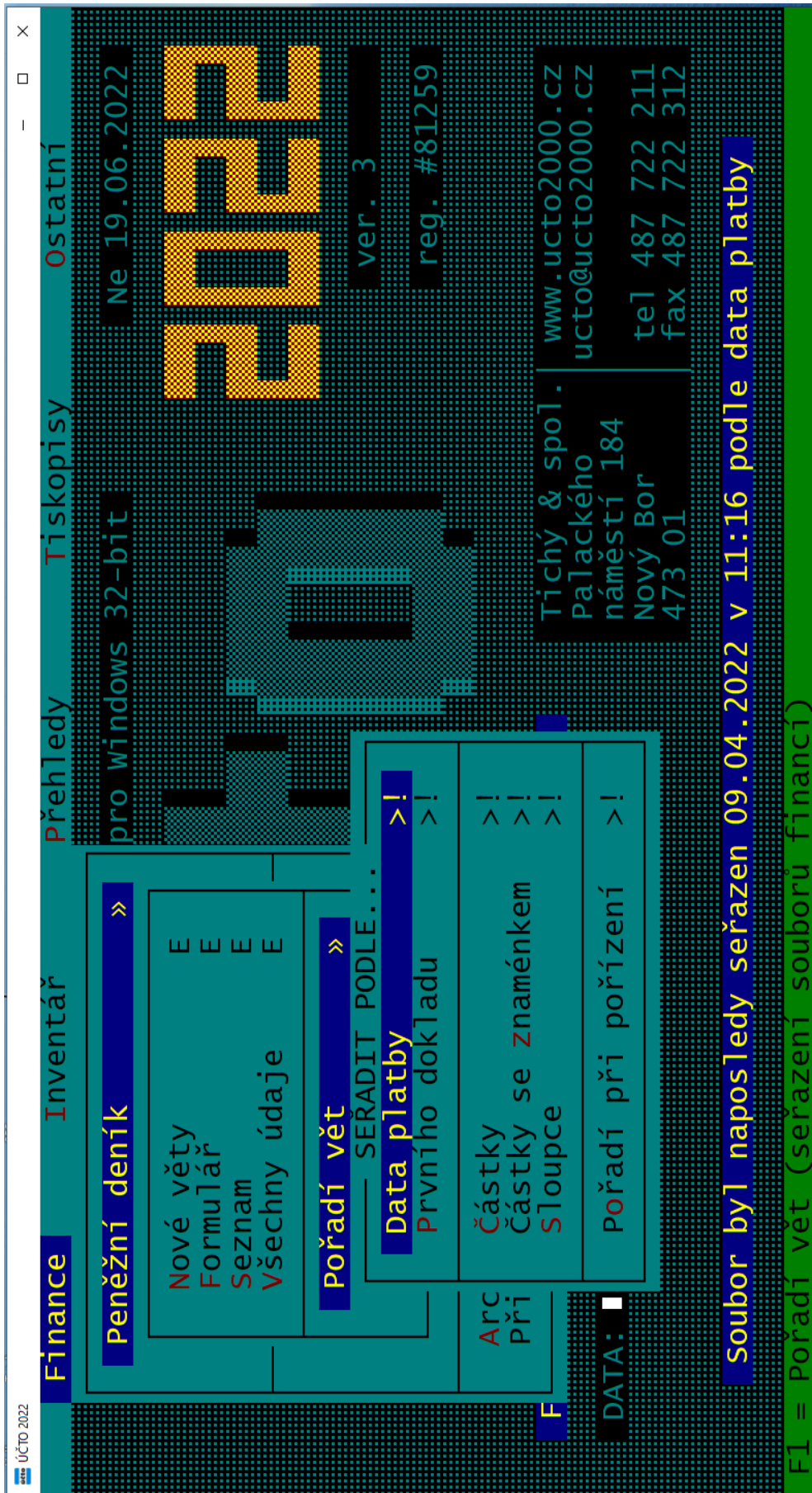


Figure 4: Sorting cannot be done on the list page

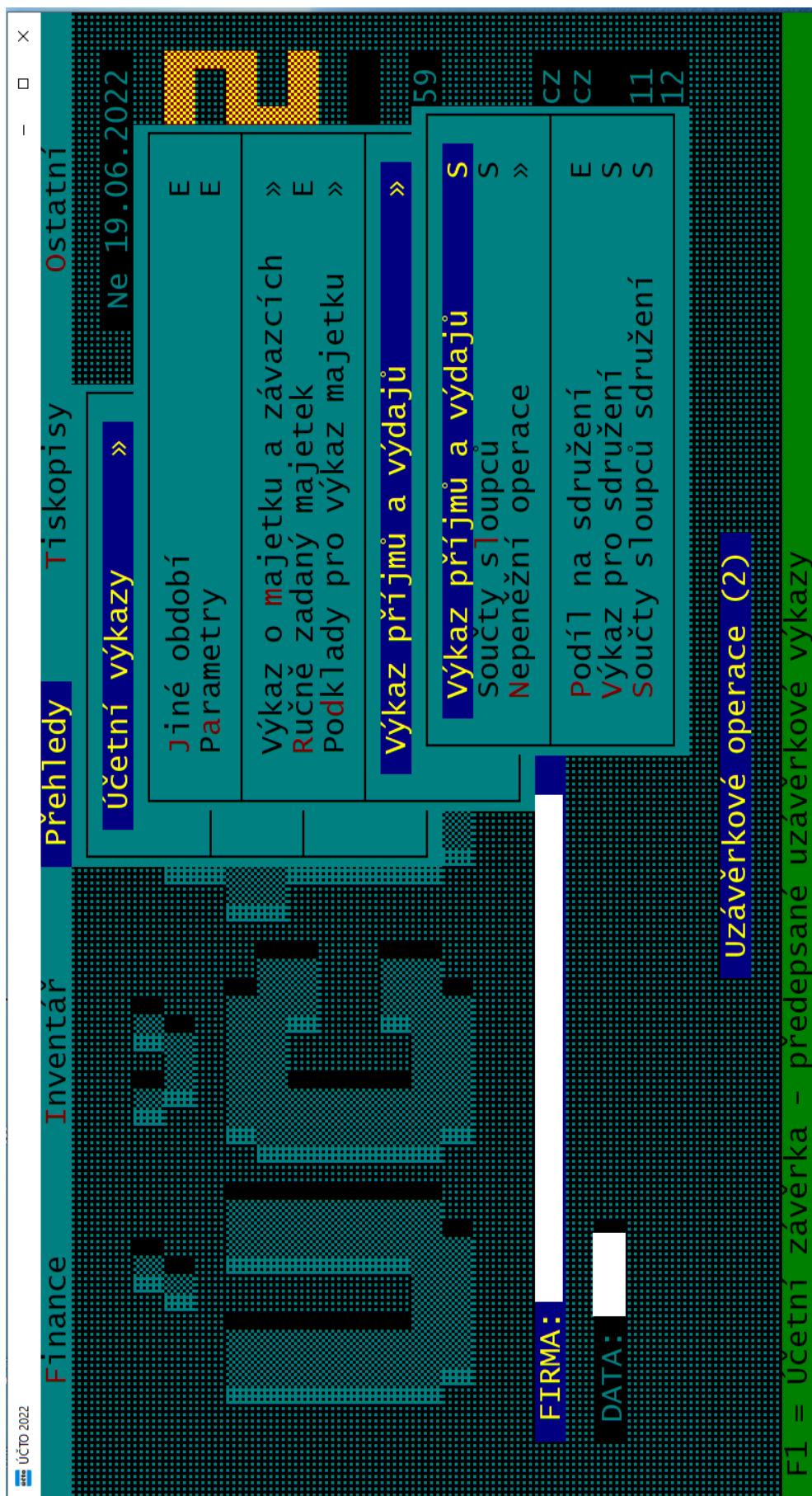


Figure 5: Overview, one of the most important functionalities, is quite hidden

E Screenshots from the application



Figure 6: Homepage

- ☰
- 🏠 Domů
- + Přidat
- Nový příjem nebo výdaj
- Nová pohledávka nebo dluh
- Nový dlouhodobý majetek
- ☰ Evidence
- 📁 Majetek
- 📄 Přehled
- 🔍 Nápověda

Nový příjem nebo výdaj

***Název:**

***Příjem nebo výdaj?**

***Číslo dokladu:**

***Vyberte datum uhrazení:**

***Částka:** Kč







***Druh úhrady:**

***Daňová položka** Ano

Drobný majetek? Ne

Popis:

Figure 7: Income or expense form

-  Domů
-  Přidat
 - Nový příjem nebo výdaj
 - Nová pohledávka nebo dluh
 - Nový dlouhodobý majetek**
-  Evidence
-  Majetek
-  Přehled
-  Náповěda

Nový dlouhodobý majetek

***Název:**

***Datum začátku používání:**

***Počáteční cena:** Kč

***Odpisová skupina:**

***Způsob odpisu:**

Popis:



 Odeslat
 Náповěda

Figure 8: Asset form

- Domů
- +
- Nový příjem nebo výdaj
- Nová pohledávka nebo dluh
- Nový dlouhodobý majetek
- Evidence
- Majetek
- Přehled
- Nápověda

Nový dlouhodobý majetek

***Název:**

***Počáteční cena:**

Popis:

Odpisová skupina
Každý dlouhodobý majetek je zařazen do odpisové skupiny. Skupina určuje způsob odpisování:

- 1 - nářadí, počítače, kancelářská technika
- 2 - nákladní i osobní automobily, pracovní stroje, traktory, autobusy
- 3 - kotle, lodě, klimatizace
- 4 - věže, budovy ze dřeva a plastů, osvětlení budov, plynovody
- 5 - byty, výrobní budovy, silnice, studny, mosty
- 6 - obchodní domy, školy, administrativní budovy, kulturní památky

Zařazování majetku může být složité. Pokud si stále nejste jisti, kam majetek zařadit, můžete zvolit skupinu 1.

Způsob odpisu
Způsob odpisu si můžete zvolit sami, podle toho, který vám více vyhovuje.

Figure 9: Hint on the asset form

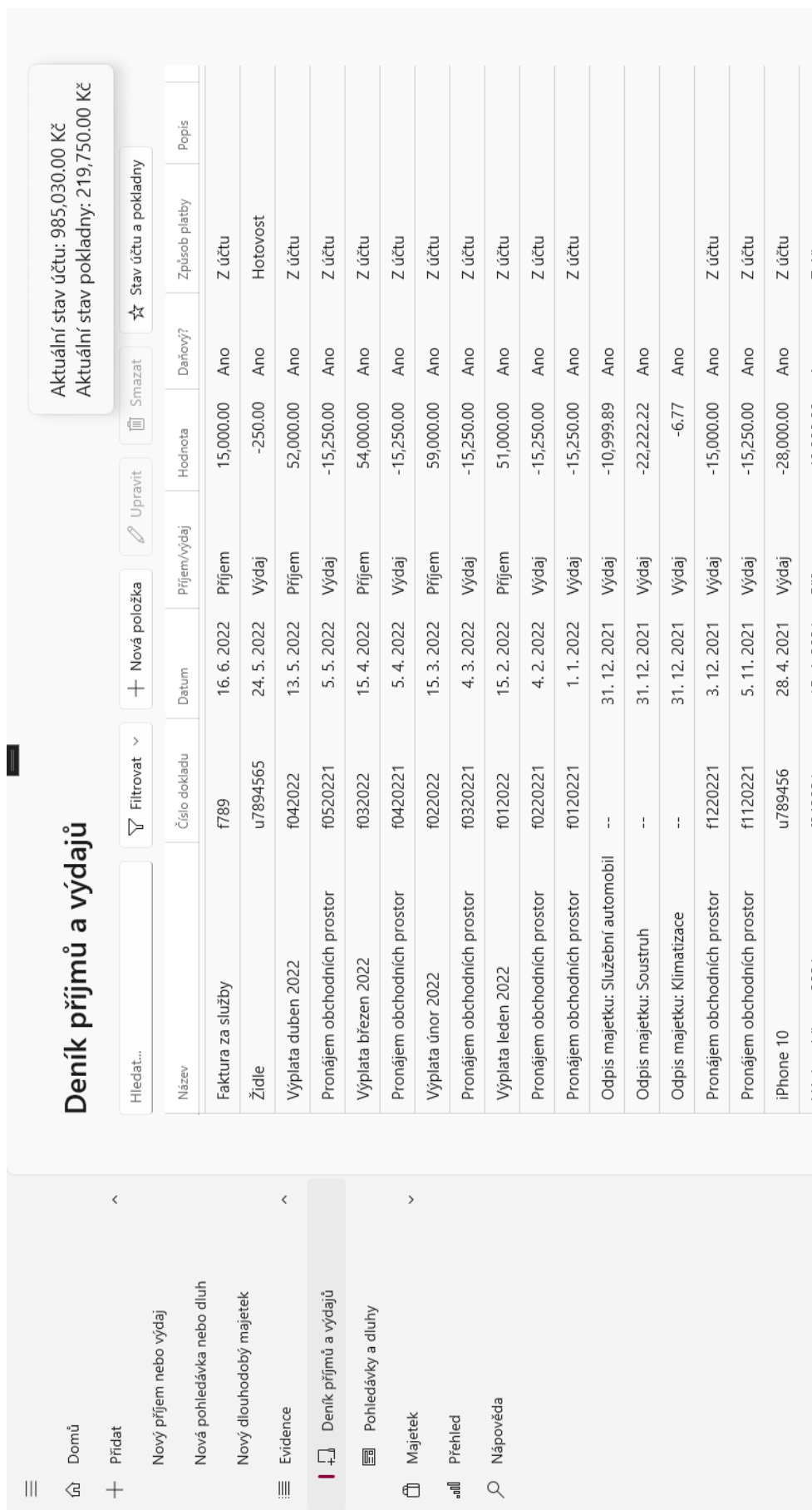


Figure 10: Income and Expense Log

- ☰
- 🏠 Domů
- + Přidat
- ☰ Evidence
- 📑 Deník příjmů a výdajů
- 🕒 Pohledávky a dluhy
- 📁 Majetek
- 📊 Přehled
- 🔍 Nápověda

Deník příjmů a výdajů

←
Editujete položku

***Název:** Pronájem obchodních prostor

***Vyberte datum uhrazení:** 5.5.2022

***Příjem nebo výdaj?** Výdaj

***Částka:** 15250 Kč

***Datová položka:** Ano

***Číslo dokladu:** f0520221

***Druh úhrady:** Z účtu

Drobný majetek? Ne

Popis: Popis nebo další informace

✓ Odeslat

↻ Vrátit změny

Figure 11: An example of an item being edited

- ☰
- 🏠 Domů
- + Přidat
- ☰ Evidence
- 📑 Deník příjmů a výdajů
- 🕒 Pohledávky a dluhy
- 📁 Majetek
- 📊 Přehled
- 🔍 Nápověda

Deník příjmů a výdajů

✖ Pozor! Číslo faktury je povinný údaj! ✕

←
Editujete položku

*Název:

*Příjem nebo výdaj?

*Číslo dokladu:

Popis:

*Vyberte datum uhrazení:

*Částka: Kč

*Druh úhrady:

*Datová položka:

Drobný majetek?

✓ Odeslat

↻ Vrátil změny

Figure 12: An example of a validation

Deník příjmů a výdajů

Filtrovat
+ Nová položka
Upravit
Smazat
☆ Stav účtu a pokladny

Název	Číslo dokladu	Datum	Příjem/výdaj	Hodnota	Daňový?	Způsob platby	Popis
Výřazení majetku: auták	--	18. 6. 2022	Výdaj	500,000.00	Ano		
Výřazení majetku: auto	--	17. 6. 2022	Výdaj	500,000.00	Ano		
Faktura za služby	f789	16. 6. 2022	Příjem	15,000.00	Ano		Z účtu
Židle			Výdaj	-250.00	Ano		Hotovost
Výplata duben 2022			Příjem	52,000.00	Ano		Z účtu
Pronájem obchodních p			Výdaj	-15,250.00	Ano		Z účtu
Výplata březen 2022			Příjem	54,000.00	Ano		Z účtu
Pronájem obchodních p			Výdaj	-15,250.00	Ano		Z účtu
Výplata únor 2022			Příjem	59,000.00	Ano		Z účtu
Pronájem obchodních prostor	f0320221	4. 3. 2022	Výdaj	-15,250.00	Ano		Z účtu
Výplata leden 2022	f012022	15. 2. 2022	Příjem	51,000.00	Ano		Z účtu
Pronájem obchodních prostor	f0220221	4. 2. 2022	Výdaj	-15,250.00	Ano		Z účtu
Pronájem obchodních prostor	f0120221	1. 1. 2022	Výdaj	-15,250.00	Ano		Z účtu
Odpis majetku: Služební automobil	--	31. 12. 2021	Výdaj	-10,999.89	Ano		
Odpis majetku: Soustruh	--	31. 12. 2021	Výdaj	-22,222.22	Ano		
Odpis majetku: Klimatizace	--	31. 12. 2021	Výdaj	-6.77	Ano		
Pronájem obchodních prostor	f1220221	3. 12. 2021	Výdaj	-15,000.00	Ano		Z účtu

Opravu chete položku smazat?

Tuto akci již nelze vzít zpět

Figure 13: An example of an item being deleted from the Income and Expense Log

Pohledávky a dluhy

Hledat... Filtrovat + Nová položka Upravit Smazat Zaznamenat platbu

Název	Číslo dokladu	Firma	Datum vytavení	Pohledávka/dluh	Hodnota	Daňový?	Popis
výdaná faktura	f/22002	Vodafone	4. 6. 2022	Pohledávka	10,000.00	Ano	Vystavená fal
Výplata květen 2020	f01052022	MůjZaměstnavatel a.s.	1. 5. 2022	Pohledávka	48,000.00	Ano	Faktura za nc
Poskytnuté služby	f08963	UdělámeWebovky.cz	30. 4. 2022	Dluh	15,000.00	Ano	Rekonstrukce
Platba za služby			11. 2021	Dluh	78,500.00	Ano	

Vyberte prosím datum zaplacení

Vyberte datum zaplacení

Odeslat Zrušit

Domů Přidat Evidence Deník příjmů a výdajů Pohledávky a dluhy Majetek Dlouhodobý majetek Drobný majetek Přehled Nápořvěda

Figure 14: Debt records, with an example of a record being marked as paid

Dlouhodobý majetek

Hledat...

Filterovat
Nový majetek
Upravit
Výřadit
Smazat

Název	Datum začátku používání	Počáteční hodnota	Zbývající hodnota	Odpisová skupina	Typ odpisu	Popis
auták	17. 6. 2022	500,000.00	0.00	2	Zrychlený	
auto	10. 6. 2022	600,000.00	0.00	2	Zrychlený	
Vrtačka	7. 1. 2022	81,000.00	81,000.00	1	Zrychlený	Vrut Extreme 300
Klimatizace	1. 7. 2021	123.00	122,986.47	3	Rovnoměrný	Klíma
Služební automobil	1. 1. 2021	99,999.00	0.00	2	Rovnoměrný	Automobil potřet
Soustruh	1. 3. 2019	100,000.00	0.00	1	Zrychlený	

Odpisy

Pořadí	Částka	Datum
1	33,333.33	31. 12. 2019
2	44,444.44	31. 12. 2020
3	22,222.22	31. 12. 2021

Figure 15: Asset records, with an example of the asset detail

Dlouhodobý majetek

Název	Datum začátku používání	Počáteční hodnota	Zbývající hodnota	Odpisová skupina	Typ odpisu	Popis
auták	17. 6. 2022	500,000.00	0.00	2	Zrychlený	
Odpisy						
Pořadí						
1						
auto				2	Zrychlený	
Vrtačka				1	Zrychlený	Vrut Extreme 300
Klimatizace				3	Rovnoměrný	Klima
Služební a				2	Rovnoměrný	Automobil potfet
Soustruh				1	Zrychlený	

Položku nelze smazat!

Na majetku již proběhly odpisy a nelze ho tedy smazat. Pokud jste majetek vyřadili, zvolte akci Vyřadit.

Figure 16: An example of an attempt to delete an asset that have already been depreciated

Dlouhodobý majetek

Hledat... Filterovat Nový majetek Upravit Výřadit Smazat

Název	Datum začátku používání	Počáteční hodnota	Zbývající hodnota	Odpisová skupina	Typ odpisu	Popis
auták	17. 6. 2022	500,000.00	0.00	2	Zrychlený	
Odpisy						
1		500,000.00				
auto			0.00	2	Zrychlený	
Vrtačka			81,000.00	1	Zrychlený	Vrut Extreme 300
Klimatizace			122,986.47	3	Rovnoměrný	Klima
Služební automobil			0.00	2	Rovnoměrný	Automobil pořifet
Soustruh			0.00	1	Zrychlený	

Vyberte důvod vyřazení:

Vyberte důvod

Odeslat Zrušit

Figure 17: An example of the act of liquidating an asset

The screenshot displays the 'Drobný majetek' (Small Assets) section of a mobile application. The interface includes a navigation menu on the left with options like 'Domů', 'Přidat', 'Evidence', 'Deník příjmů a výdajů', 'Pohledávky a dluhy', 'Majetek', 'Dlouhodobý majetek', 'Drobný majetek', 'Přehled', and 'Nápověda'. The main content area shows a table of assets with columns for 'Název', 'Hodnota', 'Datum pořízení', and 'Datum vyřazení'. The first row, 'Židle', is highlighted in red, indicating it is selected for liquidation. A modal dialog is open over the table, titled 'Vyberte prosím datum, od kdy majetek nevyužíváte' (Please select a date from when the asset is no longer used). The dialog contains a date picker labeled 'Vyberte datum vyřazení' and two buttons: 'Odeslat' (Send) and 'Zrušit' (Cancel).

Název	Hodnota	Datum pořízení	Datum vyřazení
Židle	-250,00	24. 5. 2022	
iPhone 10	-28,000,00	28. 4. 2021	17. 6. 2022
iPhone 7	-20,000,00	1. 3. 2019	28. 4. 2021
Platba za			

Figure 18: Small asset records, with an example of an asset being marked as liquidated



Figure 19: Overview for the year 2019

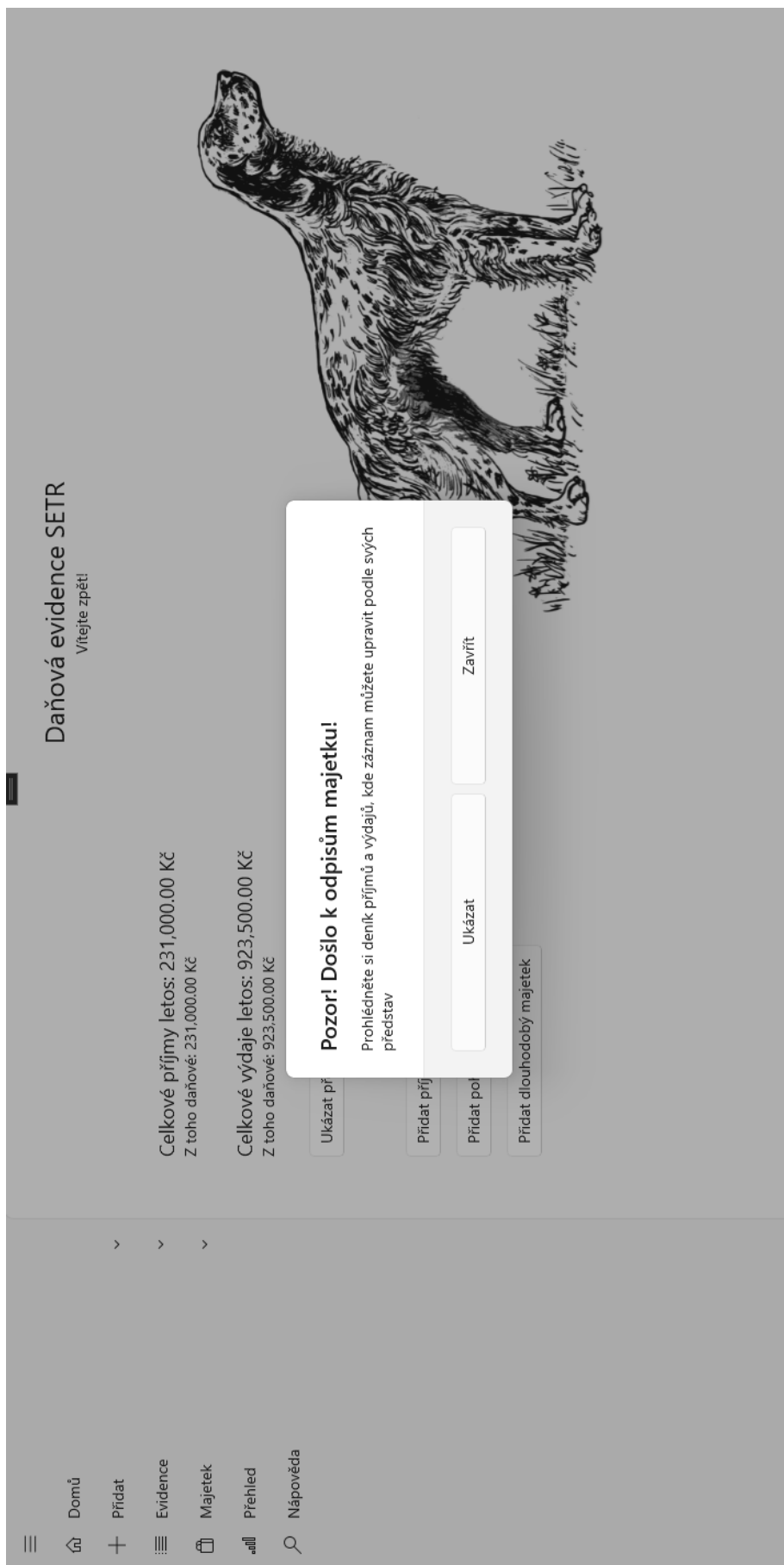


Figure 20: Depreciation popup

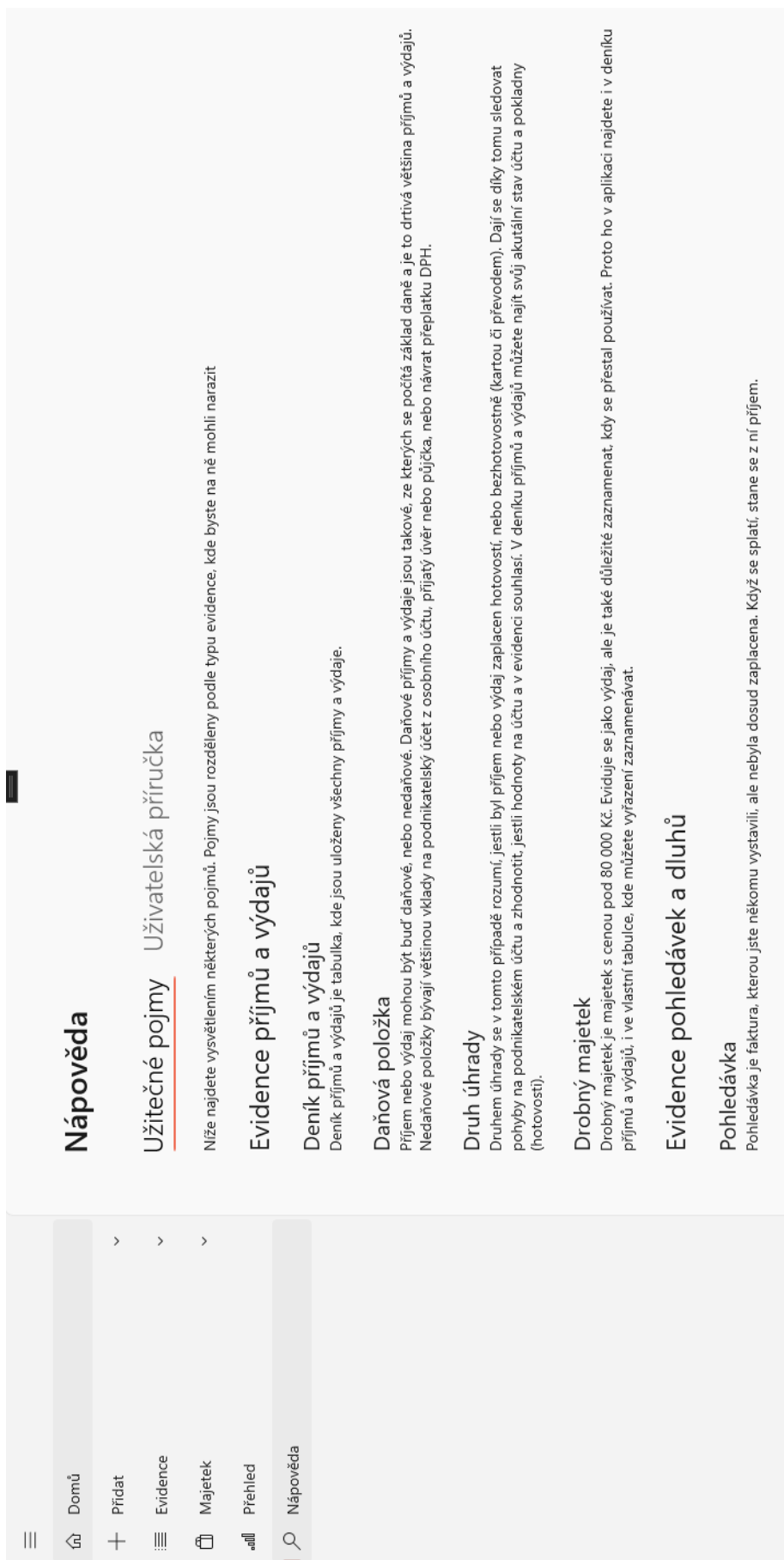


Figure 21: Hint page

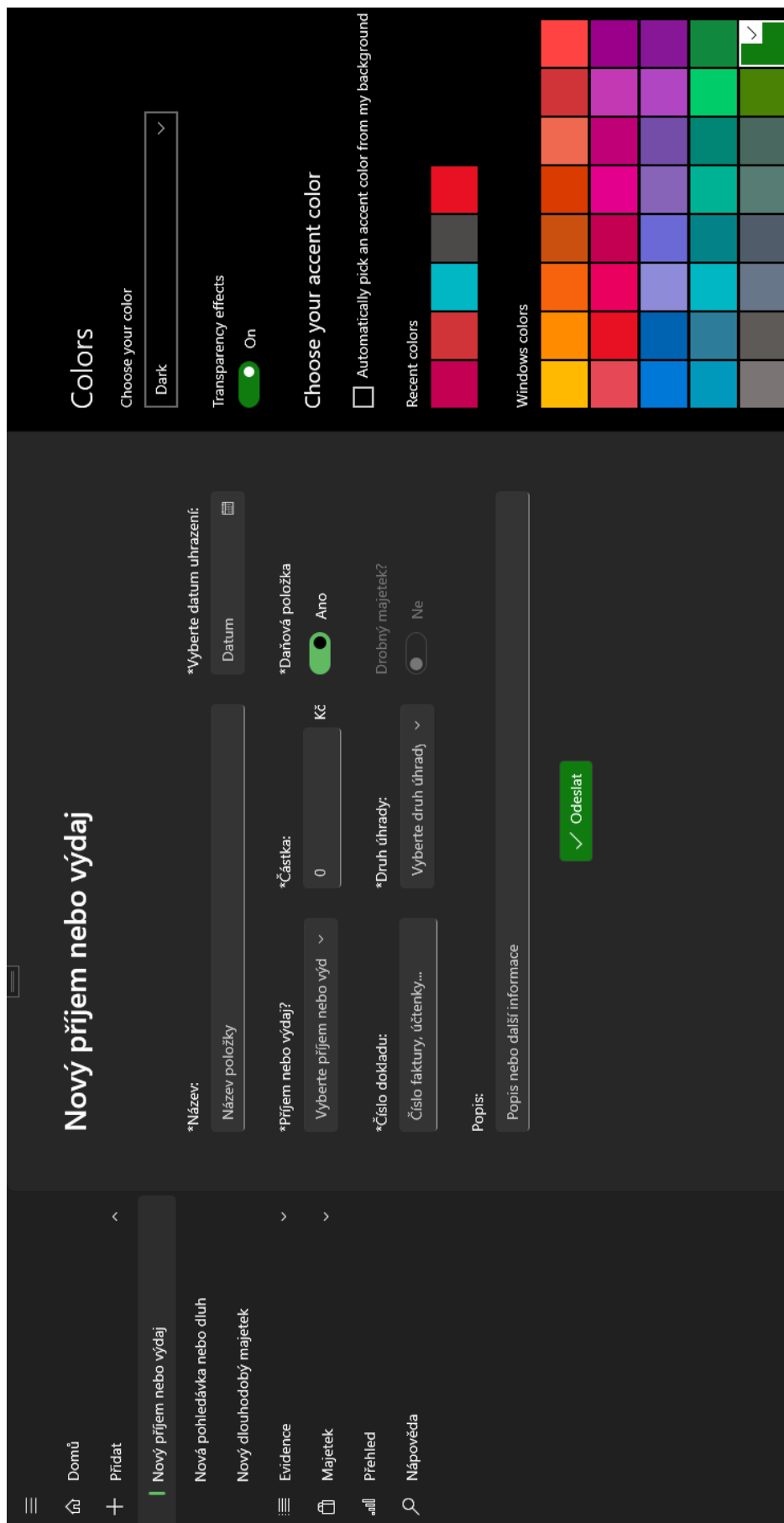


Figure 22: An example of the application being responsive to the theme and the colour set in the system

F Directory tree

```
TaxRecords
├── DBModels
│   ├── Asset.cs
│   ├── Debt.cs
│   ├── DepGroup.cs
│   ├── DepMethod.cs
│   ├── Depreciation.cs
│   └── Item.cs
├── Pages
│   ├── Assets
│   │   ├── AssetDataSource.cs
│   │   ├── AssetEditPage.xaml
│   │   │   └── AssetEditPage.xaml.cs
│   │   ├── AssetFormPage.xaml
│   │   │   └── AssetFormPage.xaml.cs
│   │   ├── AssetListPage.xaml
│   │   │   └── AssetListPage.xaml.cs
│   │   ├── SmallAssetDataSource.cs
│   │   ├── SmallAssetListPage.xaml
│   │   │   └── SmallAssetListPage.xaml.cs
│   ├── Debts
│   │   ├── DebtDataSource.cs
│   │   ├── DebtEditPage.xaml
│   │   │   └── DebtEditPage.xaml.cs
│   │   ├── DebtFormPage.xaml
│   │   │   └── DebtmFormPage.xaml.cs
│   │   ├── DebtListPage.xaml
│   │   │   └── DebtListPage.xaml.cs
│   ├── Items
│   │   ├── ItemDataSource.cs
│   │   ├── ItemEditPage.xaml
│   │   │   └── ItemEditPage.xaml.cs
│   │   ├── ItemFormPage.xaml
│   │   │   └── ItemFormPage.xaml.cs
│   │   ├── ItemListPage.xaml
│   │   │   └── ItemListPage.xaml.cs
│   ├── Overview
│   │   ├── OverviewDataSource.cs
│   │   ├── OverviewPage.xaml
│   │   │   └── OverviewPage.xaml.cs
│   ├── HintPage.xaml
│   │   └── HintPage.xaml.cs
│   ├── WelcomePage.xaml
│   │   └── WelcomePage.xaml.cs
├── App.xaml
│   └── App.xaml.cs
└── AppLogic.cs
```



```
|
├─ Datasource.cs
├─ DBContext.cs
├─ MainWindow.xaml
└─ MainWindow.xaml.cs
```