FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Aplikace pro řízení rizik kybernetické bezpečnosti
Student:	Julia Plotnikova
Vedoucí:	Ing. Jiří Dostál, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

Řízení kybernetických bezpečnostních rizik (dále jen „rizik") je nedílnou součástí zavádění a provozování ISMS (Information Security Management System) v každé organizaci. Této skutečnosti si je vědoma i EU, která v nové směrnici NIS 2 klade mimo jiné důraz i na řízení rizik. Zmiňovaná směrnice NIS 2 výrazným způsobem rozšiřuje počet povinných subjektů, které se musí řízení rizik věnovat. Pro řadu organizací to bude povinnost nová a bude to pro ně znamenat návrh a zavedení nových procesů. Renomované společnosti nabízejí softwarová řešení pro řízení rizik jako součást svých řešení a služeb. Pro menší a střední společnosti však budou taková řešení finančně nedostupná nebo prakticky nepoužitelná (např. OneTrust GRC nebo Comply RISK).

Cílem práce je navrhnout procesy pro řízení kybernetických bezpečnostních rizik v menších podnicích a vytvořit software (webovou aplikaci), která bude tyto procesy implementovat a podporovat.

1) Seznamte se s prostředím malého či středního podniku s ohledem na kybernetickou bezpečnost.
2) Analyzujte možnosti, dostupnost a praktickou použitelnost softwarových produktů pro řízení rizik.
3) Na základě potřeb organizace navrhněte procesy pro řízení rizik.
4) Navrhněte a vytvořte prototyp softwarového produktu pro řízení rizik.
5) Otestujte prototyp v reálném prostředí.

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 14. ledna 2023 v Praze.

Bakalářská práce

APLIKACE PRO ŘÍZENÍ RIZIK KYBERNETICKÉ BEZPEČNOSTI

Julia Plotnikova

Fakulta informačních technologií
Katedra informační bezpečnosti
Vedoucí: Ing. Jiří Dostál Ph.D.
11. května 2023

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů kódu

Chtěla bych poděkovat především vedoucímu Ing. Jiří Dostálovi Ph.D. za jeho rady, odborné vedení a podporu při psaní této práce. Dále bych chtěla vyjádřit poděkování své rodině, kamarádům a příteli, kde díky jejich emocionální podpoře bylo zpracování bakalářské práce snažší.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.
Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 11. května 2023 .....

Abstrakt

Bakalářská práce na téma „Aplikace pro řízení rizik kybernetické bezpečnosti“ se v první části zabývá analýzou řízení rizik pro menší až střední podniky. Práce se zde věnuje různým prostředím a podnímkám rizik, jaké jednotlivé kroky se musí splnit a jak nejlépe ohodnotit tato rizika. V další části se práce zabývá analýzou aplikace pro řízení rizik a následně implementace a programování prototypu a jak jej implementovat. V poslední části byl prottyp otestován v reálném prostředí, abychom zjistili, kde má tento prototyp problémy.

Klíčová slova webová aplikace, prototyp, kybernetická bezpečnost, řízení rizik, NIS 2

Abstract

The Bachelor's thesis on „Aplikace pro řízení rizik kybernetické bezpečnosti“ addresses risk management analysis for small to medium-sized enterprises in the first part. The thesis discusses various environments and conditions for risks, the individual steps that must be taken, and how to best evaluate these risks. The second part of the thesis deals with an analysis of the risk management application, followed by the implementation and programming of a prototype and how to implement it. In the last section the prototype was tested in a real-world environment to identify any problems our prototype may have.

Keywords web application, prototype, cybersecurity, risk management, NIS 2

Seznam zkratek

CRR	Cyber risk register
GRC	Governance, Risk management and compliance software
ISMS	Information security management system
RMT	Risk management team

Kapitola 1
Úvod

Bob: „Alice, slyšela jsi někdy o řízení rizik?“
Alice: „Ano, o řízení rizik jsem slyšela. Je to důležitá součást zavádění a provozování systémů pro správu kybernetické bezpečnosti v organizacích. Mám to správně?“
Bob: „Naprosto.“

Kybernetická bezpečnost je důležitým tématem v současném digitálním světě. Pro organizace je klíčové podporovat ochranu dat a zabezpečení informačních systémů. Zavádění procesů pro řízení kybernetických rizik je proto nutným krokem pro každou organizaci, která chce ochránit své informace a minimalizovat rizika spojená s kybernetickými útoky.

Výsledkem této práce bude prototyp, navrhnutý pro menší a střední podniky, který jim ve finální fázi umožní řídit kybernetická rizika a minimalizovat jejich dopad na organizaci. Vytvořením webové aplikace, která bude implementovat navržené procesy pro řízení rizik, bude organizacím poskytnuta ucelená a snadno použitelná řešení.

Téma bylo vybráno s ohledem na novou směrnici NIS 2, která klade důraz na řízení rizik a rozšiřuje počet povinných subjektů. Vzhledem k tomu, že pro menší a střední podniky mohou být dostupná řešení finančně nedostupná, bylo rozhodnuto vytvořit nové řešení, které bude bezplatné a snadno použitelné pro menší až střední organizace.

1.1 Cíl práce

Cílem bakalářské práce je navrhnout procesy pro řízení kybernetických rizik v podnicích, navrhnout a vytvořit prototyp, který bude tyto procesy spravovat. Prvním krokem je seznámení se s prostředím malého či středního podniku s ohledem na kybernetickou bezpečnost. Tato část bude zaměřena na zjištění specifických vlastností této skupiny a na identifikaci hlavních rizik, se kterými se potýkají.

Dále bude provedena analýza existujících softwarových produktů pro řízení rizik. Tato část bude mít za cíl posoudit, zda jsou dostupné a použitelné nástroje, které by mohly být využity v podnicích, a jak by mohly být tyto nástroje využity.

Na základě potřeb organizace budou navrženy procesy pro řízení rizik. Tato část bude zahrnovat vytvoření postupů pro identifikaci, hodnocení a řízení rizik. Tyto procesy budou navrženy s ohledem na potřeby a specifika podniků.

Po navržení procesů bude vytvořen prototyp softwarového produktu pro řízení rizik. Tato část bude zahrnovat návrh architektury softwaru a implementaci klíčových funkcionalit. Prototyp bude vytvořen s ohledem na potřeby menších a středních podniků. Prototyp bude obsahovat primární funkce pro podporu řízení rizik v podnicích.

Poslední částí bude otestování prototypu v reálném prostředí. Tato část bude zahrnovat testování

funkcionality v reálném provozu a posouzení jeho účinnosti v řízení rizik v podnicích. Výsledky testování budou zaznamenány a z toho bude vyvozen závěr, kde prototyp selhal a kde prototyp splnil potřeby podniku.

1.2 Stručný obsah kapitol

Rešerše Zde si projdeme zdroje a získáme potřebné znalosti, abychom mohli vytvářet procesy na řízení rizik a následně i aplikaci na řízení rizik.

Analýza řízení rizik Zde si navrhneme procesy okolo řízení rizik a definujeme si potřebné znalosti. Dále vytvoříme procesy pro jednotlivá rizika.

Analýza aplikace na řízení rizik Zde si navrhneme od samotného začátku aplikaci na řízení rizik.

Testování Otestujeme aplikaci na skutečných datech a zda vše vyhovuje potřebám podniku, popřípadně které body by více vyhovovaly.

Závěr Na závěr si řekneme, které cíle jsme splnili, které nesplnili a proč tomu tak bylo.

Kapitola 2
Rešerše

Alice: „Bobe, vytvořila jsem ti výjimku ve firewallu, nebude z toho problém?“
Bob: „Je to dočasná výjimka, pak nesmíme zapomenout tu výjimku zrušit.“
O týden později jak Alice tak i Bob zapomenou na tuto výjimku, protože ji nikam nezapsali.

2.1 Úvodní definice

Jelikož se bakalářská práce bude zabývat kybernetickými riziky, je vhodné pro správné pochopení problematiky si zavést definice:

Hrozba (threat) Příčina incidentu zneužívající zranitelnosti, kde tento incident může způsobit narušení důvěrnosti, integrity a dostupnosti systémů v podniku.
Příklad hrozby: Napadení webového serveru prostřednictvím posílání hodně dotazů z různých stanic, způsobujíc denial of service [1].

Zranitelnost (vulnerability) Vlastnost systému či aplikace, kterou útočník může potenciálně zneužít k vlastnímu zisku či k porušení tzv. CIA triády[1].
Příklad zranitelnosti – systém pro uživatelích vyžaduje jednoduchá hesla.

Riziko (risk) Možnost ztráty/škody primárních či podpůrných aktiv podniku[1].

Denial of service (DoS) Dopad na dostupnost, systém je přehlcen požadavky a dokud je nezpracuje, tak se jeví jako nedostupný. Nemusí jít vždy o útok.

CIA triáda „Confidentiality, Integrity, Availability“, 3 vlastnosti popisující bezpečný systém [2].

- **Confidentiality** (*Důvěrnost*) Informace nebudou dostupná nepovolaným osobám.
  Příklad narušení důvěrnosti: kreditní údaje klientů jsou viditelné na hlavní stránce webové stránky
- **Integrity** (*Integrita*) Informace poskytované systémem jsou důvěryhodná a není možné s informacemi neoprávněně manipulovat. Tedy informace mají být ve výsledku autentické, přesné a důvěryhondé.
  Příklad narušení integrity – útočník se bude vydávat jako důvěryhodný server a my pak budeme pracovat s daty, které on nám dodal a které nejsou správně.
- **Availability** (*Dostupnost*) Osoby s přístupem budou moct přistupovat k informacím, tehdy, kdy potřebují.
  Příklad narušení dostupnosti – Studenti o půlnoci zjistí, že se jim otevřel rozvrh a zahltí požadavky webový server a kvůli tomu přestane odpovídat, než stihne dokončit a odpovědět na ostatní požadavky.

Aktivum (asset) Vlastnictví podniku, které má pro tento podnik hodnotu. Dělí se na primární a podpůrná aktiva nebo také na hmotná a nehmotná aktiva. O primární aktivum se jedná v případě, jestli to podnik používá jako hlavní příjem financí. V opačném případě – podpůrné aktivum – slouží jako nápomocná složka, aby podnik mohl fungovat. Mějme fiktivní firmu, která vyrábí metalické polotovary. Primární aktiva by byly jednotlivé stroje, které ten daný polotovar vyrábějí a podpůrné aktivum by byly například servery, díky kterým můžou poskytovat zákazníkům informace o svých výrobcích[1].

Událost § 7 (1) Kybernetickou bezpečnostní událostí je událost, která může způsobit narušení bezpečnosti informací v informačních systémech nebo narušení bezpečnosti služeb anebo bezpečnosti a integrity sítí elektronických komunikací [1]

Incident § 7 (2) Kybernetickým bezpečnostním incidentem je narušení bezpečnosti informací v informačních systémech nebo narušení bezpečnosti služeb anebo bezpečnosti a integrity sítí elektronických komunikací v důsledku kybernetické bezpečnostní události. [2]

V jiných zdrojích literatury je možné se setkat s pojmem **ISMS** – „Information security management system“, jedná se tedy o nástroj pro správu informační bezpečnosti v podnicích. Správa rizik je zahrnuta v těchto systémech, ale ISMS a CRR (Cybersecurity risk register) nejsou zaměnitelné pojmy, ISMS je širší pojem a řízení rizik je nedílnou součástí takového programu.
Správa rizik zahrnuje analýzu, jaké incidenty můžou nastat a co může být důsledkem těchto incidentů. Dále, v závislosti na této informaci se navrhne a schválí reakce na riziko a do kdy se tato reakce na riziko má uplatnit. Taková reakce buď nemá žádný následek, nebo mitiguje riziko, tj. snižuje buď dopad či pravděpodobnost vzniku incidentu.

2.2 Situace

Představme si menší či střední podnik (takže například do 1000 zaměstnanců pro střední podnik), která spravuje a provozuje počítačovou síť routerů a dalších síťových zařízení, aplikace zakoupené od třetích stran a další nástroje, umožňující zaměstnancům podniku pracovat na produktu.

V určitém okamžiku se podnik setká s kybernetickými zranitelnostmi a pokusí se je v lepším případě evidovat a řešit například s pomocí Excel souboru. Toto řešení může být efektivní v některých případech, například v malých firmách. Nicméně, jakmile se firma začne rozrůstat, stává se tento soubor nezvladatelným a přístup vícero uživatelů je problematický. Další problém představuje reakce vedení, které v některých případech nemusí chápat hrozbu plynoucí z rizika a nevěnuje tomu dostatečné zdroje (finanční nebo usilí zaměstnanců). Taková situace může vést k mnohem větší ztrátě, než by stála oprava problému. Proto je pro menší firmy vhodné zvolit lepší náhradu pro řešení kybernetických rizik.

Z toho plyne, že menší firmy by měly ideálně od začátku používat software na řízení rizik.

2.3 Nástroje na řízení rizik

Hlavním cílem aplikace pro řízení rizik je **efektivně evidovat, klasifikovat, reagovat a spravovat rizika**. Tato aplikace by měla být flexibilní vůči počtu uživatelů. Pro začátek by postačil návrh s pomocí tabulek v Excelu či podobném tabulkovém procesoru.

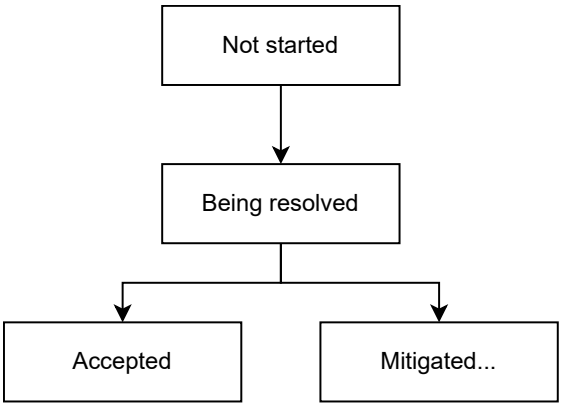
[1] Zákon 181/2014 – Zákon o kybernetické bezpečnosti
[2] Zákon 181/2014 – Zákon o kybernetické bezpečnosti

Tabulka 2.1 Ukázka evidence rizik v menším podniku

Risk name	Responsible person	Status
Špatná verifikace v databázi	Bob	Mitigated
Chybí pravidlo v firewallu	Bob	Accepted
Detekované AP se stejným názvem	Alice	Not started
Chybné nastavení switche	Charlie	Being resolved

Mezi jednotlivé předpoklady řízení rizik u tohoto podniku patří mimo jiné i to, že rizika evidujeme anglicky, u rizika jenom evidujeme *název, zodpovědnou osobu a stav* a že daná množina stavů je následující[3]:

Obrázek 2.1 Stavy rizik u menšího podniku



Malé firmy se budou často snažit najít rychlé a levné řešení pro správu rizik, aby úspěšně řešily akutní problémy s minimálními náklady a úsilím. I v případě, že tyto firmy budou mít nástroj na řízení rizik, je pravděpodobné, že nebude mít stejnou funkcionalitu, jako placené nástroje. Daná fiktivní firma v případě řešení rizik by pravděpodobně zavedla těchto pár stavů.

K jednotlivým stavům se dostaneme blíže později.

V našem modelu tohoto fiktivního podniku se předpokládá, že podnik se stará o bezpečnost aktiv a tím pádem se taky stará o rizika. U menších firem existují tendence nedbat na informační bezpečnost a minimalizovat náklady.

Jak se nám tato firma postupně rozrůstá, tak se časem zvyšuje zájem o zavedení lepšího nástroje na správu rizik. Tedy k nám do naší firmy nastoupí dalších X lidí, díky čemuž se nacházíme v mezistupni malé a střední firmy.

V předchozím modelu řízení rizik byla schválně vynechána jedna kategorie a nyní se podíváme, proč tomu tak bylo.

Touto chybějící položkou je závažnost rizika. V případě malé firmy by se to mělo komunikovat, co je potřeba vyřešit a soustředit se na ostatní věci, kritické k existenci samotné firmy.
Závažnost je problematická, protože je velmi objektivní. Už při menší analýze webových stránek, vysvětlujících jak nakládat s riziky, tak se zřídkakdy shodli na jednotné škále. Od například 1-3 škály, nebo i 1-99% škála, kde nižší číslo znamená menší dopad. Existují zde i jiné typy škálování rizik, textové, od „low“ (nízké) po „high“ (vysoké) či „critical“ (kritické). Je to na firmě, které škálování si vybere. Více o této problematice později při navrhování procesů pro rizika.
U menší firmy předpokládejme škálu od 1-10. Nyní můžeme mluvit o komplexnějších procesech, kde ideálně zodpovědná osoba bude určovat číslo reprezentující závažnost, tak má smysl i mluvit o **dopadu** a **pravděpodobnosti**. Na těchto dvou hodnotách by pak měla záviset závažnost rizika (například vynásobení $impact * probability$ je v tomto případě ideální).

[3] Všechny diagramy v této práci jsou vytvořené s pomocí nástroje app.diagrams.net (draw.io).

Přidejme hodnocení závažnosti rizika:

Tabulka 2.2 Návrh tabulky pro řízení rizik v pětičlenném podniku

ID	Risk name	Responsible person	Severity	Status
623	Poor verification in database	Bob	10	Mitigated
926	Missing rule in firewall	Bob	5	Accepted
1820	Detected AP with the same name	Alice	3	Not started
2390	Incorrect switch configuration	Alice	8	Being resolved

V případě firmy je velmi potřebné si nastavit škálu, tak, aby vyhovovala osobám posuzujícím rizika, jelikož se například může stát, že Bob ohodnotí riziko s identifikačním číslem 2390 jako *8*, ale Alice by riziku 2390 dala hodnocení *8.9*. Obojí hodnocení jsou správně, ale nejsou objektivní. Naopak omezit to na škálu 1-3 by mohlo znamenat i ztrátu závažnosti rizika a s tím přehlédnutí některých rizik, které jsou závažnější a měly by být řešeny dřív. Tím pádem jsem pro tento případ zvolila hodnocení jako $i \in \{1,2,...,10\}$, kde se předpokládá, že jak dopad tak pravděpodobnost bylo vzat na vědomí. V aplikaci/prototypu pak budu používat hodnocení rizik jako $i \in \{1,2,...,5\}$, popřípadně i jiné hodnocení .

Komplexnějším hodnocením rizik se budu věnovat později v textu. Jedná se o například CVSS a CWSS. Popíšeme si je detailněji a následně je i budeme částečně využívat v prototypu.

Jako další informaci, která by se hodila většímu podniku, je **deadline** pro vyřešení rizika, tj. do kterého data se musí vyřešit riziko (tedy mitigace apod., o tom si víc povíme později) respektive se posunout do dalšího stádia.

2.3.1 Ukázka tabulky ze skutečného podniku

Jedná se již o větší podnik se zaměřením na kybernetickou bezpečnost. V rámci anonymizace jsem vynechala název tohoto podniku, jelikož toto nejsou volně dostupné informace.

Definice CRR v tomto podniku:

> Risk Register: A risk register is a risk management tool that can assist in creating a repository of potential risks that could affect the organization and its operations. It describes the nature and severity of each risk, possible mitigation measures and risk „owners“ While the risk register can be a helpful compliance tool, its purpose is best fulfilled when used alongside other risk management frameworks, tools, organizational cultural risk awareness, stakeholder buy-in and accountability. This tool was created by taking and adapting some of the toolkits that a number of the INGOs in this study use as part of their risk management practice.

Používá se zde škálování od 1 do 5 pro pravděpodobnost a dopad.

- pravděpodobnost **1**: událost může nastat pouze v extrémních situacích ještě jednou za 100 let
- pravděpodobnost **2**: událost se nestala v minulosti, může se stát ještě jednou za několik let
- pravděpodobnost **3**: stala se v minulosti, může se stát ještě jednou za rok
- pravděpodobnost **4**: možná událost ještě jednou za pár týdnů či měsíců
- pravděpodobnost **5**: častá událost ještě jednou za pár dnů či týdnů

Co se týče dopadů, tak je můžeme rozdělit do několika různých typů:

- Provozní dopad – například zpoždění během provozu či vývoje
- Finanční dopad – podnik bude(může) mít při naplnění rizika peněžitou ztrátu

- Dopad na reputaci podniku – podnik může být terčem negativních médií a pozornosti veřejnosti
- Dopad na zaměstnance – například zveřejnění osobních informací zaměstnanců
- Dopad na kulturu podniku – zaměstnanci zpochybňují politiku/motivaci/management, rezignují z podniku
- Jiné druhy dopadů

Jednotlivé úrovně dopadů se pak kategorizují dle finančního dopadu, kde hodnota závisí na intervalu ztráty, tato hodnota se odvíjí od zisků a podobných údajů.

2.3.1.1 Jejich risk matrix

Tato excel tabulka obsahuje i definici „risk matrix“ (respektive „riziková matice“, ale název ponechám v češtině). Reprezentuje skupiny rizik.

	Impact				
Likelihood	**Negligible**	**Minor**	**Moderate**	**Severe**	**Critical**
Very likely	Low	Medium	High	Very high	Unacceptable
Likely	Low	Medium	High	High	Very high
Moderately likely	Very low	Low	Medium	High	High
Unlikely	Very low	Low	Low	Medium	Medium
Very unlikely	Very low	Very low	Very low	Low	Low

2.3.2 Excel tabulka na řízení rizik

Poznamenám, že tato firma již nepatří mezi malé až střední podniky, ale už se jedná o relativně velkou firmu. Jejich nástroj na řízení rizik (excel soubor) vypadá následovně: Hlavička obsahuje tyto položky (kde jsou závorky typu tak to označuje dropdown, tj. možnost výběru z nabídky):

- Úroveň: {Země, Podnik, Oblast}
- Department: {Komunikace, Plnění a kontrola předpisů, Finance a účetnictví, Zdraví a bezpečnost, IT, Právní oddělení, Logistika, Operace, Bezpečnost}
- Datum vytvoření
- Datum posouzení

Po hlavičce následují rizika:

- ID rizika
- Typ rizika {Fiduciární, Bezpečnostní, Zdraví a bezpečnost, Reputační, Plnění předpisů, Právní, Informační bezpečnost, . . . }
- Kategorie rizika {Zastupování, Odezva na katastrofy, Externí komunikace, Granty, Lidské zdroje . . . }
- Popis rizika
- Dopad (1 až 5)
- Pravděpodobnost (1 až 5)
- Hodnocení rizika (Dopad x Pravděpodobnost)
- Dopad na cíle

- Strategie mitigace rizik (Vypište opatření, která zabrání nebo sníží šanci na výskyt rizik)
- Výsledky mitigace rizik (Ohodnoťte účinnost zmírnění)
- Dopad (po mitigaci)
- Pravděpodobnosdt (po mitigaci)
- Jedná se o přijatelné riziko?
- Vlastník rizika (garant)
- Zodpovědný ředitel (vlastník řízení rizik)
- Časový plán (očekávané datum dokončení položky)

2.4 O aplikacích na řízení rizik

CRR je Cyber risk register, neboli taky Cyber risk management application (CRMA), obojí jsou názvy, které slouží k označování aplikace řízení rizik.

Před návrhem analýzy rizik je vhodné si shrnout podstatné vlastnosti takového CRR, které jsme si již ukázali a ty, které na první pohled nemusí být zřejmé.

Organizace potřebují informační systémy, aby mohly poskytovat službu či dále provozovat byznys. Tento úspěch je podmíněn tím, že CIA triáda zůstane chráněná. Samotná CRR aplikace toho není schopná, avšak je efektivním nástrojem k tomu, aby se ochránila CIA triáda, v případě, že se používá podle toho, jak to bylo zamýšleno (viz dále).

CRR by neměl sloužit jako hlavní nástroj pro vytváření dokumentace či dalšího papírování, ale převážně jako nástroj na řešení a vizualizaci/zdůraznění naléhavých problémů, které je potřebné vyřešit co nejdříve.

Dále je potřebné, aby všichni zodpovědní byli zapojení do risk registru a věděli o dění v podniku. Toho lze docílit dobrým uživatelským designem a zároveň také tím, že uživatelé budou informováni o nových událostech v risk registru, například, pokud riziko změní status na nový, tak by jistá skupina lidí (garant, delegát apod.) by měli dostat email, či jakýkoliv jiný druh upozornění.

2.5 Existující aplikace

V této části se budeme zabývat existujícími řešeními. Hlavním cílem této části je dokumentovat, co již existuje. Cílem této bakalářské práce není vytvářet již existující aplikaci, ale spíše vytvořit novou aplikaci pro menší podniky (konkrétně prototyp aplikace).

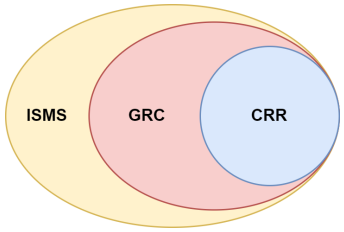
Není cílem zde porovnávat finanční náklady na tento software, jelikož je to high-end software cílený na větší podniky, tím pádem budou víc nákladné. Pro účely této práce nás zajímají funkcionality týkající se zaznamenávání a řízení rizik, jiné aspekty těchto programů zde nebudou analyzovány.

Je také nutné podotknout, že tyto programy jsou komerční, kdežto výsledkem této bakalářské práce bude prototyp, který nemá mít komerční využití, nýbrž bude volně dostupný.

K těmto produktům jsem nedostala DEMO verzi, budu parafrázovat obsah z stránek daného produktu. Taktéž i cena produktu není volně přístupná.

GRC nástroj jedná se o další „název“ pro aplikace pro řízení rizik. Zkratka znamená „Governance, Risk management and compliance software“, kde řízení rizik je jednom jednou ze tří součástí. Je nutné si to neplést s ISMS, ISMS je nadstavbou i k GRC a pracuje s informační bezpečností.

Obrázek 2.2 Grafické znázornění spojitosti ISMS, GRC a CRR.



OneTrust GRC OneTrust GRC [3] nabízí hodně nástrojů pro organizace na řízení rizik a kontrolu dodržování předpisů. Podporují primárně cloud-based strukturu – díky tomu nabízejí centralizované místo pro změnu a monitorování informací (jak rizik tak i dodržování předpisů).

Platforma nabízí různé moduly pro spravování různých částí řízení rizik, jako například „Vendor risk management“ [4], „incident and breach management“ [5] a spravování politik pro řízení rizik.

Také nabízí integraci s jinými platformami a aplikacemi.

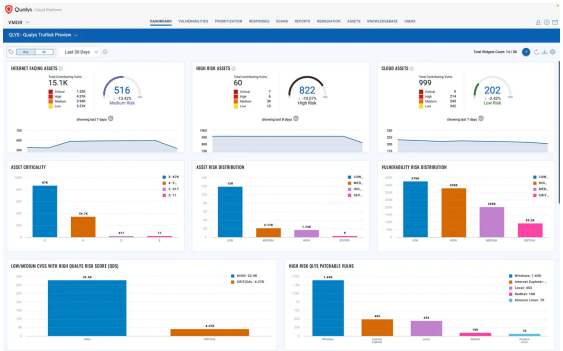
Lze zde poznat, že tento software míří na větší firmy, které se musí řídit více specifickými nařízeními, protože podporuje dodržování předpisů jako například GDPR, CCPA a HIPPA.

Qualys vulnerability management solution Jak je evidentní z názvu, jedná se o produkt od firmy Qualys ([4]). Aplikace pracuje s pojmem „vulnerability“ (zranitelnostech), ale zabývá se stejnou problematikou jako řízení rizik.

Program obsahuje Dashboard s oddělením na otevřené/opravené/ignorované zranitelnosti. Také obsahuje listování zranitelností, má vlastní způsob na vyhodnocování závažnosti zranitelnosti (tj. dají se používat vlastní vzorečky). Podle toho vzorečku pak přiřadí hodnotu rizika (např. Critical Risk).

Ostatní funkce slouží hlavně kvůli podpoře zranitelností jako například skenování, mapování zařízen.

Obrázek 2.3 Dashboard Qualys vulnerability management solution



SailPoint Access Risk management Produkt SailPoint Access Risk management ([5]) je navržen pro větší firmy. Je navržený pro řízení rizik, granulární visibilitu. Cloud-based a automatizované řízení přístupu, díky tomu pak vzniká kontinuální analýza rizik. Umí automaticky generovat auditní reporty. Systém je škálovatelný a adaptabilní vůči požadavkům firmy.

[4] Tuto část raději přenechám v angličtině. Nejbližší překlad do češtiny je „Řízení rizik prodejce/dodavatele“, jedná se o rizika týkající se případů, kdy dodavatel (jiná firma či entita) selže.

[5] Zde podobně jako v předchozí poznámce pod čarou, překlad do češtiny není nejlepší, jako breach se zde jedná o porušení jedné ze složek CIA triády, konkrétně důvěrnosti. Nejbližší český překlad by byl „porušení“, ale nenese to se sebou daný kontext.

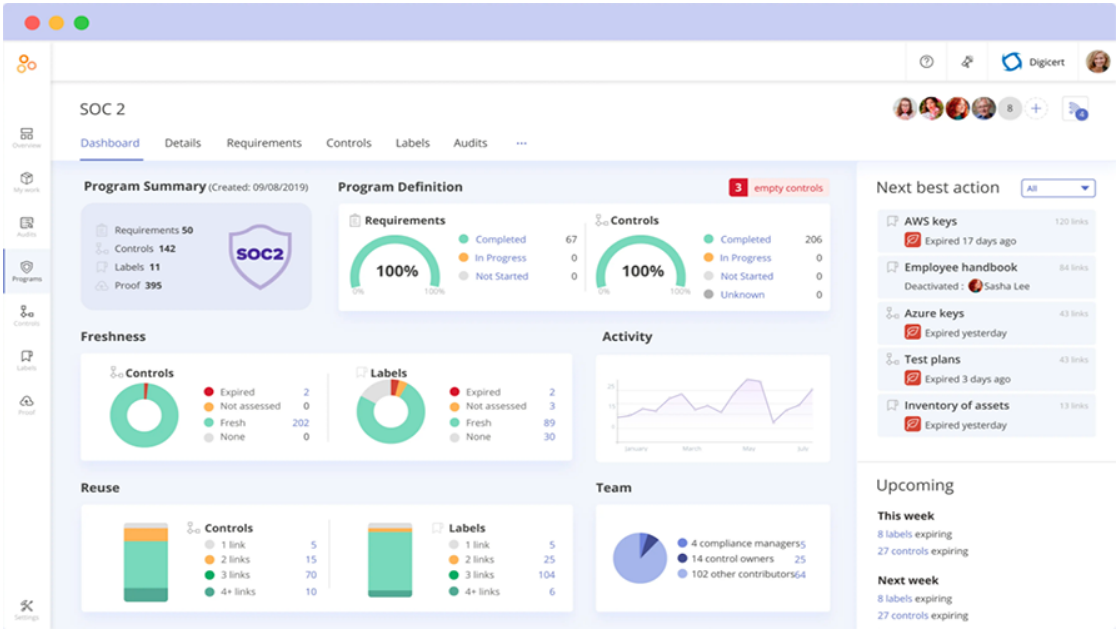
Jenom autorizované osoby mají přístup k citlivým informacím a systémům.

Hyperproof GRC Hyperproof GRC ([6]) je navržen pro velké organizace a pomáhá organizacím dodržovat přepisů a standardů, jako jsou HIPAA, GDPR a PCI-DSS.

Obsahuje centralizovaný řídící panel, kde se zobrazuje stav dodržovaných předpisů a také například automatizovaných workflows.

Nabízejí speciální „compliance fabric“, což je kolekce předem vytvořených frameworků [6] a standardů.

Obrázek 2.4 Hyperproof GRC



[6] Zde taktéž, překlad „framework“ je „rámec“, je zde nutné se naklonit anglické variantě, jelikož česká varianta tohoto slova ztrácí význam v tomto kontextu.

Kapitola 3

Analýza řízení rizik

Bob: „Zjistil sem, že politika nastavení hesel uživatelských účtů není úplně správně nastavená, uživatel musí mít minimálně heslo o pěti znacích.“
Alice: „Tím pádem někteří můžou mít slabá hesla?“
Bob: „Ano, musíme si to nějak poznamenat, ať se to vyřeší.“

3.1 Příprava pro řízení rizik

Před tím, než se začneme zabývat analýzou rizik, musíme definovat několik dalších klíčových pojmů:

Garant Osoba, která je zodpovědná za riziko (resp. rizika) - rozhoduje o tom, zda řešení rizika odpovídá směrnicím v podniku.

Delegát Osoba, která zastupuje garanta. Má stejné pravomoci jako garant.

Risk management team (RMT) (dále jako RMT) Skupina zaměstnanců nebo i jednotlivec, který je zodpovědný evidenci a řešení rizik. Riziko pak řeší s garantem či týmem, který má vliv na toto riziko, aby se implementovala správná mitigace atd.
Hlavním cílem RMT je ochránit podnik nákladově efektivním způsobem Aby práce RMT byla optimalizovaná, musí se splnit tyto podmínky [7]:

- Musí být nastavená hodnota akceptace rizika seniorním managementem
- Existují zdokumentované procesy a procedury řízení rizik (k procesům si povíme něco později a jeden takový si popíšeme)
- Existují procesy pro identifikaci a mitigaci rizik (spravuje RMT)
- Dostačující zdroje či finance zařízené seniorním managementem
- Cybersecurity-awareness trénink pro všechny zaměstnance (nutné pro uživatele pracujícím s počítačem, dobrovolné pro ostatní)
- Možnost zorganizovat týmy pro výpomoc v problematických místech (při výskytu akutního problému)
- Mít zmapované legální a regulační dodržování požadavků
- Vývoj metrik a výkonových indikátorů pro měření a správu rizik
- Schopnost identifikace a posouzení nových rizik při změnách prostředí a firmy

3.1.1 Identifikace rozsahu

Vyřeší se granularita systému – tj. stanovení konkrétních informací, které budou evaluovány, a jakým způsobem bude tato evaluace provedena, tedy jak malé detaily budou zahrnuty a jak budou celky rozděleny.

Představme si firmu, která má oddělení na rozvoj aplikací, kde se tyto aplikace budou používat v rámci podniku. Pak takový podnik má možnost tyto jednotlivá aktiva navázat na rizika (tj. rozdělění podle např. oddělení, jednotlivé aplikace, projekty atd.).

Firma si v daném případě může vyřešit aktiva jako například tak, že za každý projekt se „vytvoří“ 1 aktivum, ke kterému se budou vázat jednotlivá rizika. Vedle těchto projektových aktiv se vytvoří i jiná „podniková“ aktiva, ke kterým zaměstnanci budou evidovat jiná rizika.

3.1.2 Chápání úrovně akceptovatelného rizika

V rámci RMT se musí nastavit jednodné chápání akceptovatelného rizika a taky chápání škály rizik, tj. jaké dopady jsou „horší“ (způsobují větší škodu) než ta zbylá rizika. Toto se musí řešit kontinuálně.

Ideálně hodnotu závažnosti rizika provázat s peněžitou ztrátou, respektive udělat různé finanční kategorie v závislosti na ztrátě a podle toho navazovat hodnotu.

Dle normy ISO/IEC 27005 se jedná o tzv. „Context establishment“ – kdy pro jednotlivá rizika vytvoříme kontext, aby RMT byl schopný určit závažnost rizika [8].

3.1.3 Použití metod na identifikaci rizik

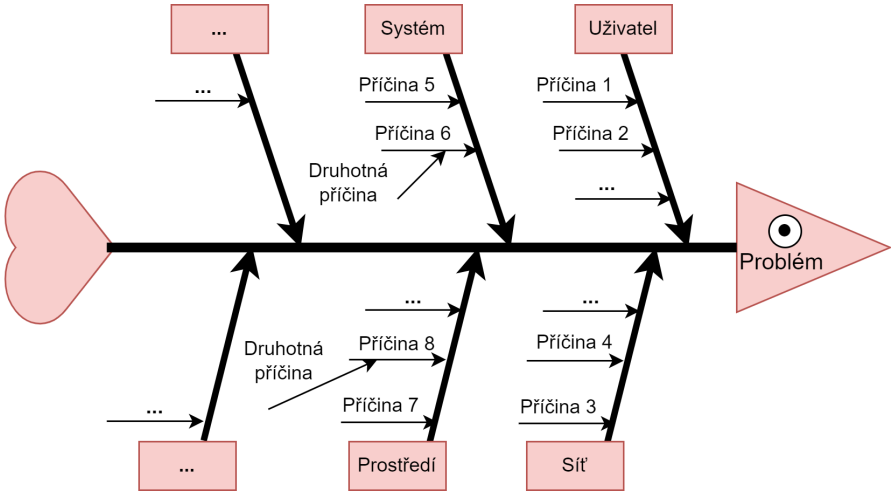
Delphi metoda Informace o rizicích získáme anonymně a s předem danou strukturou (například s pomocí dotazníku). Zde RMT má jako povinnost tyto informace evidovat a zpracovat je dále. Po zpracování vytvoří jednotlivá rizika a zde zodpovědná osoba (například garant aktiva, ke kterému riziko náleží) sumarizuje rizika a toto jde podkladem pro další kolo. V dalším kole se proces opakuje (avšak nevytváří se další rizika, jenom pozměnují). Proces končí tehdy, kdy rizika přijatá od anonymních uživatelů jsou totožná. Nakonec se rizika uloží do systému [9].

Brainstorming Založení meetingu či organizování společné setkání zaměstnanců, kteří pracují na daném aktivu/mají zkušenosti s danou tématikou apod. Na této poradě zaměstnanci spolu vymýšlejí a zpracovávají jednotlivá rizika společně [9].
Tato metoda je nejpřímočařejší a nejjednodušší.

Root cause analysis Jedná se o způsob identifikace faktorů a příčin – případě, že máme incident, snažíme se přijít na důvod toho, proč se to stalo, namísto toho, abychom jednoduše řešili důsledky.
Jednotlivé způsoby na nalezení příčiny[9]:

- **5 WHY** Pětkrát se zeptáme, proč k tomuto incidentu došlo. Na konci těchto pěti otázek bychom měli dostat příčinu.
- **Cause and effect diagram** Například diagramy typu „Ishikawa“ neboli „fishbone“. Nakreslí se rybí kostra směřující hlavou doprava. Hlava této ryby reprezentuje problém, který v daný moment řešíme. Na páteř kostry jsou napojené různé příčiny daného problému, ty jsou znázorněné jako jednotlivá žebra kostry. Pro všechny tyto příčiny se také doplňují vodorovně přispívající faktory.

Obrázek 3.1 Ukázka Ishikawa diagramu



Existují další způsoby pro nalezení příčiny rizika, ale zmiňuji se zde o těch nejjednodušších.

Inspekce Neboli audity, probíhají buď přes automatizované nástroje, testování nebo kontaktování lidí v týmu. Je zde potřebné vytvořit checklist informací, které musí daný projekt/aktivum splňovat. Cílem je identifikovat a zamezit incidentům, které nejsou v souladu se standardy.

3.1.4 Nutné kroky pro jednotlivá aktiva

1. Identifikace aktiv
   - Rozdělení na primární a podpůrná aktiva, v případě potřeby i jemnější rozdělení, například podle projektů. Je možné toto rozdělení mít jinak, například hmotná a nehmotná aktiva. Toto závisí na podniku.
   - Pod které oddělení patří, které oddělení se stará o toto aktivum (v případě třetí strany firma), kdo je zodpovědný za správnou funkcionalitu tohoto aktiva.
2. Posouzení hodnot těchto aktiv
   Můžeme si položit otázky: o kolik peněž přijdeme, když se tomuto aktivu něco stane? Jak hodně závážná tato aktiva jsou pro náš podnik? Tuto hodnotu by měl primárně určovat garant a manažer, respektive zaměstnanci, kteří mají dobrý přehled o podniku a ostatních aktivách.

3.1.5 Proces pro jednotlivá rizika

Tento proces rozdělíme na jednotlivých pár stavů, na kterých pak budeme stavět dále (zde si o nich něco málo řekneme a dále si to rozvijeme):

1. **Identifikace rizika**
   Jako první bod musíme nalézt rizika. Tedy se provádí identifikace všech událostí, které mají negativní důsledek na jednu z vlastností triády – důvěrnost, integrita a dostupnost.
   Pro každé riziko se najde 1 garant (případně delegát), později zodpovědný za posouzení, ošetření či monitorování rizika.
   Riziko by musí mít krátký název popisující, v čem je problém (popisující, v čem je to riziko unikátní). Pak by také mělo mít popis („Pokud *'riziko'* tak se stane *'následek rizika'*.“) Riziko by mělo mít přiřazeno datum, do kdy se musí posoudit (to má na starosti RMT).

2. **Analýza a posouzení rizika**
   Riziko by dále mělo mít nějaké finální datum, do kterého by to mělo být vyřešeno (uvedeno do akceptovatelného stavu).

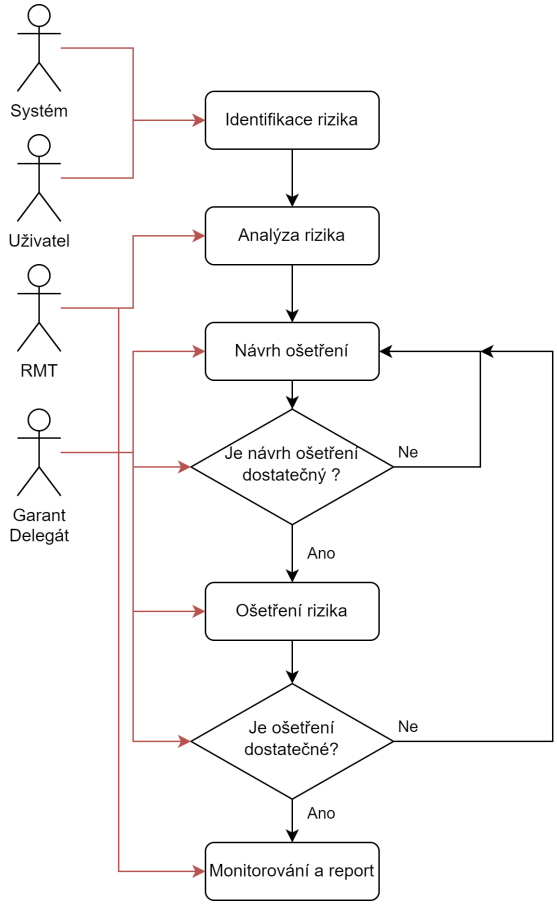
3. **Ošetření rizika**
   Neboli mitigace rizika. Jedná se o snížení dopadu či pravděpodobnosti rizika.

4. **Monitorování a report rizika**
   Monitorování rizika v porovnání s předchozí variantou rizika bez mitigace.
   V této fázi se také nastavuje datum, do kdy se toto riziko musí znovu zkontrolovat, což může být například rok, do kdy se to musí kontrolovat, pak je toto riziko zavedené do systému jako nové riziko, nebo v závislosti na nastavení podniku se toto riziko nastaví jako znovu aktivní a bude se muset znovu kontrolovat a posuzovat [10].

Dále následuje dělení těchto jednotlivých částí řízení rizik a také více informací k nim.

Při celém procesu řízení rizik je nutné, aby všechny zodpovědné osoby (manažeři, RMT a osoby zodpovědné za provoz) komunikovali a byli si vědomi změn a ošetření rizik a měli možnost kontaktovat jiné osoby, jestliže jim něco nepřijde jako vhodné řešení.

Obrázek 3.2 Diagram popisující průběh rizika



3.1.6 Identifikace rizika

Smyslem identifikace rizika určit co se stane a jaký to bude mít dopad na podnik. Také je účelem získat přehled o tom, jak, kde a proč by mohla finanční ztráta nastat.

Je nutné shromaždovat všechna rizika, i ta, která nejsou pod kontrolou podniku.

Uživatel nebo systém najde nějakou zranitelnost a vytvoří tak zde informaci o tom. Jedná se o nalezení rizika, ale také o nalezení dopadu tohoto rizika. Riziko se musí navázat na nějaké akvtivum, aby samotné riziko mělo smysl. V případě systému (nebo nějakého importu) se toto navázání může dít automaticky a v případě uživatele a v závislosti na jeho znalostech firmy či znalosti oboru buď uživatel sám naváže aktivum, nebo to za něj propojí někdo z RMT či lidí majících s tím zkušenosti.

3.1.7 Analýza a posouzení rizika

Tuto část má na starosti RMT.
Pod tuto kategorii spadá, jak už z názvu vyplývá, analýza rizika. Analýzou rizik se myslí přiřazení hodnoty rizika dle toho, jaký systém hodnocení RMT či jiná zodpovědná entita zvolila. Pod analýzu také spadá analýza kontextu daného rizika, například mějme stejné riziko v nějakém softwaru, nějaký port je otevřený tam, kde není žádná jiná ochrana a software je napojený v prvním případě na kritickou službu a v druhém případě na testovací server. Se stejným rizikem máme velmi odlišné úrovně dopadu na podnik. K tomu je potřebné znát kontext.

Pro hodnocení rizika je zde tato část:Hodnocení závažnosti rizik.

3.1.8 Ošetření rizika

Tuto část má na starosti garant rizika. Garant zodpovídá za to, že se s rizikem naloží tak, aby to bylo pro podnik dostačující. Ošetření rizika neboli **mitigace** rizika spočívá v tom, že zmírníme dopad nebo zmenšíme pravděpodobnost rizika. Jsou rizika, u kterých není možné snížit dopad či pravděpodobnost na nulu, například mějme hesla. U hesel v případě slabé politity (jenom minimum 4 znaků, bez omezení na ostatní symboly jako čísla, velká a malá písmena či speciální symboly) je toto velké riziko, a při zavedení silnější politiky [1] je lámání hesel stále možné, nezbavíme se rizika, ale je to o dost méně pravděpodobné, že se útočníkovi podaří takový útok.

Ošetření rizika se musí schválit garantem (nebo delegátem), pokud dle garanta je ošetření rizika nedostatečné, tak se navrhuje další možné ošetření rizika.

Mitigace nástává jako vhodným řešením rizika v tom případě, že náklady na vyřešení mitigace jsou menší jak ztráty. V opačném případě by se mělo zvolit **Akceptování rizika**.

Garant navrhne, jak se bude riziko mitigovat (nebo jestli se bude akceptovat/přenášet). RMT to garantovi schválí a může se postoupit do další fáze nebo to garantovi zamítne s nějakým odůvodněním. Garant musí náležitě reagovat na zamítnutí a nabídnout nové řešení.

3.1.8.1 Vyhnutí se riziku

V některých případech se lze kompletně vyhnout riziku. Jedná se také o mitigaci, ale po této mitigaci nezůstává zbytkové riziko.

3.1.9 Akceptování rizika

V případě, že jsou náklady na mitigaci rizika příliš vysoké (porovnáváme se spojeným aktivem), pak je pro podnik lepší akceptovat riziko. Ne vždy se ale jedná o finančí ztrátu, může se také jednat o ztrátu

[1] zde si dovolím poznamenat to, že zde je nejlepší se řídit doporučením NIST SP 800-63B-3, kde se limituje jenom na minimální počet znaků a nevnucuje se tam podmínka toho, jestli se tam mají vkládat speciální charaktery, protože to pak snižuje entropii uživatelských hesel, zjednodušujíc možný útok na hesla.

reputace firmy. V tomto případě je vhodné, aby se na tomto rozhodnutí podílelo vícero zodpovědných zaměstnanců firmy. Toto se netýká jenom Akceptování rizika, ale i jiných „řešení“ rizik.

3.1.10 Přenesení rizika

Neboli transfer. Jedná se pojištění u třetí strany nebo rozdělení možných újmů s jinou firmou.

> ISO-IEC 27005: „Sharing with another party the burden of loss or benefit of gain for a risk“. [8]

neboli v překladu: „Sdílení zátěže ztráty nebo přínosu zisku za riziko s jinou stranou“

3.1.11 Monitorování a report rizika

Už máme aplikovanou mitigaci(vyhnutí se)/akceptanci/přenesení rizika. V této fázi kontrolujeme, jestli naše řešení bylo dostatečně efektivní (tím je myšleno, jestli má očekávaný účinek). V případě, že to nemá dostatečný účinek, tak se musí vrátit ke kroku s mitigací a implementovat novou mitigaci. Tento cyklus se opakuje dokud není efekt dostačující.

V pozitivním případě označíme riziko jako vyřešené, vytvoří se report tohoto rizika a datum příští revize. V době této revize se musí riziko znovu otevřít a znovu se spustí proces řízení rizik od identifikace rizika. V případě že se již nejedná o riziko, lze jej odstanit ze seznamu.

3.1.12 Hodnocení závažnosti rizik

Jak již bylo poznamenáno, tato část představuje jedno z hlavních problémů na rizicích – schopnost objektivně posoudit závažnost rizika. Běžné faktory ve firmě:

- Finance potřebné na odstranění rizika
- Úroveň klasifikace dotčeného aktiva
- Porušení bezpečnosti informací (porušení CIA triády)
- Narušení chodu firmy (anglicky „operations“) – interní nebo třetí strany
- Ztráta obchodní nebo finanční hodnoty
- Porušení právních, regulačních nebo smluvních poplatků

Existují i jiné faktory, firma si může vytvořit další faktory. Pak na základě toho, jak riziko může potenciálně narušit tyto faktory, tak by měl dostat skóre.

3.1.12.1 CVSS a CWSS

CVSS (Common vulnerability scoring system) CVSS je standardní framework pro hodnocení závažnosti zranitelností (v našem případě i rizik), který byl publikován NIST v roce 2005. Výsledkem CVSS je hodnocení od 0 do 10, přičemž vyšší hodnocení značí závažnější zranitelnosti. Hodnocení CVSS je založeno na několika faktorech, včetně možnosti útočníka ohledně access[2] k této zranitelnosti , její exploatability, dopadu a možnostem protiopatření k mitigaci [11].

Hodnocení CVSS je vypočítáno na základě vzorce, který zohledňuje různé faktory a přiděluje jim hodnotu. Například zranitelnost, která je snadno zneužitelná a může být použita k získání plné kontroly nad systémem, by získala vyšší hodnocení než zranitelnost, která je obtížně zneužitelná a má omezený dopad.

[2] Tato část se překládá špatně do češtiny, ale hodnotí se, v jaké „části“ sítě se útočník musí nacházet, aby jeho útok byl úspěšný. Například útok z lokální sítě nebo například útok z TOR browseru.

Základní dělení CVSS je na „Base“ (základní), „Temporal“ (dočasné), „Environmental“ (prostředí, neboli dopad na podnik). Ke každé kategorii je pak přiřazeno několik vlastností, na základě kterých se pak počítá finální hodnocení, například pro „Base“ je zde „Availability Impact“ a hodnocení tohoto faktoru je „None“, „Low“ a „High“.

Nyní je CVSS ve verzi 3.1 a online kalkulačka od společnosti NIST je dostupná zde.

CWSS (Common weakness scoring system) je systém, který se používá k posouzení závažnosti slabých míst v softwaru, narozdíl od CVSS, kde se zaměřujeme na zranitelnosti. Byl publikován v roce 2014 společností MITRE[12].

CWSS poskytuje číselné hodnocení na stupnici od 0 do 100, přičemž vyšší hodnocení značí závažnější problém. Hodnocení je založeno na několika faktorech, včetně pravděpodobnosti, že bude slabé místo zneužitelné, potenciální dopad zneužití slabého místa a snadnosti mitigace.

Podobně jako u CVSS je hodnocení CWSS vypočítáno na základě vzorce, který zohledňuje různé faktory a přiděluje jim hodnotu.

Základní dělění CWSS je na „Base“ (základní), „Attack surface“ (neboli z jakých míst může útočník úspěšně zaútočit) a „Environmental“ (prostředí, neboli dopad na podnik).

Online kalkulačka je zde.

Kapitola 4

Analýza aplikace

Alice: „Když teď víme o tom riziku, co s ním uděláme?“
Bob: „Schováme to za tamtu kytku“
[7]

V této kapitole se dostaneme k tomu, jak napsat program na řízení rizik a jak na to jít. Už máme zázemí co se týče rizik, už víme, proč a jaká data u rizika evidujeme, co nás zajímá a jak s nimi nakládáme, popřípadně kdo a jak s nimi nakládá. Nyní už je čas na vytvoření toho nástroje, je pro nás jako uživatele výhodnější pracovat s prohlížečovou aplikací místo jednotlivým položkám v databázi či je mít v nespočet souborech ve sdíleném prostředí či Excelu.

Jelikož se jedná o prototyp, zvolila jsem jiný druh vývoje aplikace, kde budu postupně iterovat. Důvodem je to, že jsem selhávala v návrhu databáze a rozhodla jsem se pro návrh, kde v každé iteraci přidám další feature, které zde pak zdokumentuji a ve finále bude shrnutí databázového modelu a jednotlivých stránek.

Finální nástroj by pak byl založený na tomto prototypu.

4.1 Výběr technologie

Já budu používat Blazor server jako framework pro webovou aplikaci, ale je možné použít i libovolný jiný webový framework, jako například PHP a další jiné frameworky.

Důvodem tomu je, že Blazor je nová technologie a chtěla jsem se s touto technologií lépe seznámit. Obsahuje taky „nativně“ prevence útoku typu SQL injection (s pomocí jazyku C# lze použít parametrizaci v kódu) a Cross site scripting [1]. Co se týče jiných bodů zranitelnosti aplikace (jestliže se držíme OWASP Top Ten 2021), pak je ještě například problematický broken access control. Stejně jako u ostatních webových zranitelností, broken access control záleží jenom na programátorovi, žádný framework tomuto není schopen předejít. Později si ukážeme, jak můžeme omezit na základě přihlášení či rolí to, co vidí uživatel (pro Blazor).

Poznamenám do budoucna, budu používat Microsoft SQL Server (express, tedy lokální server). Je možné použít libovolnou databázi, ale v tomto momentě si programátor/analytik budou muset dávat pozor na správné datové typy a dostatečně dlouhé datové typy (v případě nvarchar nebo varbinary). Já jsem tyto políčka ponechala prázdná, jelikož se od nich implementace moc neodvíjí.

[1] Je zde nutné si pamatovat, že tyto útoky lze potenciálně umožnit, ale pokud se programátor drží zavedených pravidel, není to pro útočníka lehkým úkolem napáchat škodu

4.2 Počátky

4.2.1 Příběh

Máme malou fiktivní firmu „CodeBytes“, která vyvíjí software. V této firmě 3 zaměstnanci pracují na řízení rizik, mimo jinou práci – Alice, Bob a Charlie. Budeme při analýze aplikace vycházet z této fiktivní firmy.

Přidávání a správa uživatelů není zadáním bakalářské práce, avšak pro funkcionalitu samotného programu či demonstrace je potřebná. Pokud by podnik chtěl použít tento program pro řízení rizik, počítá se s tím, že si bude muset sám vyřešit provázání aplikace se systémem na uživatele. Jelikož jsou uživatelé stežejní část, začnu s návrhem pro uživatele.

Jako první je nutné si navrhnout uživatele, jestli to bude na základě Windows účtu v podniku, aplikace bude řešit přihlašování lokálně (přihlásíte se v aplikaci) nebo pokud v podniku existuje SSO [2].

V případě, že jsme již navrhli uživatelský systém (tj. jestli víme, s jakou technologií budeme uživatele přihlašovat do systému), pak pokračujeme na další část ohledně uživatelů – rozdělení na role. Zde je nutné vědět, který ze dvou přístupů je lepší pro daný podnik (toto by měl určit bezpečnostní manažer či jiná osoba zodpovědná za vývoj aplikace):

- N uživatelů a 1 role
- N uživatelů a M skupin

Rozdíl mezi těmito dvěma skupinami je ve složitosti. V první skupině může být uživatel například buď zaměstnanec, bezpečnostní analytik nebo admin. Podle toho dostane roli. Tato role bude pak nadále sloužit k autorizaci. V případě, že je uživatel admin, může všechno, v případě, že uživatel je bezpečnostní analytik tak může jenom importovat rizika.

Na opačné straně zde leží N:M vazba mezi uživateli a rolemi, respektive skupinami. Toto rozdělení je taktéž možné, ale je těžší na implementaci. Dovoluje lepší manipulaci s uživateli a funguje jako maska. Skupiny by měly být v jistém pořadí, aby nedocházelo kolizím s právy. Vybrala bych ternární hodnocení povolit, zdědit, odmítnout nebo další možné reprezentace těchto hodnot. V případě, že v uživatelských rolích ta nejvyšší hodnota vyhrává. Pokud má uživatel pouze „zdědit“ ve všech skupinách, tak se mu přístup zamítne.

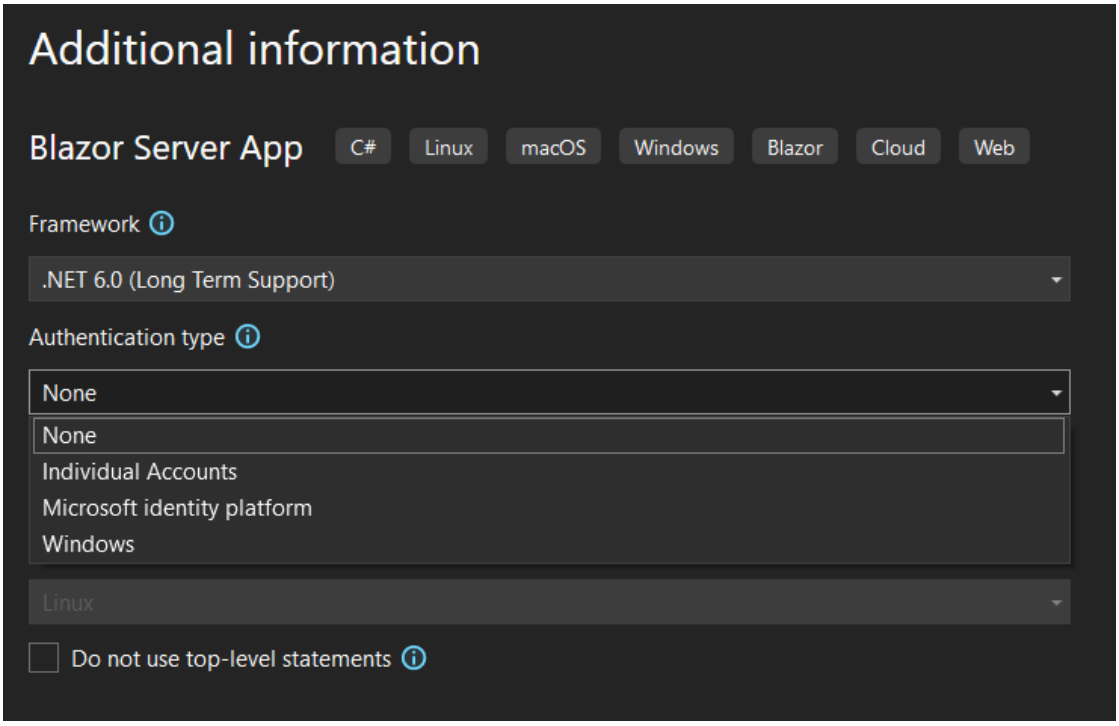
Já jsem si zvolila N:1 kardinalitu, jelikož je to prototypová aplikace a toto řešení je více než postačující.

4.2.2 Moje implementace uživatelů

Jelikož cílem tohoto prototypu je jednoduché řízení rizik, tak používání Active directory či jiné podobné služby nebudu řešit a tato část se ponechává pro firmy, které toto řešení by v budoucnu chtěli používat. Mnou zvolená technologie – Blazor serverside application nabízí několik různých typů autentizace.

[2] Single Sign On, studenti a učitelé z ČVUT s tím mají zkušenosti. Je to portál na přihlášení, který vrací token uživatele, což je kód, kterým se uživatel identifikuje a který obsahuje informace o jeho oprávnění ke službám, ke kterým má přístup. Tento token se pak obvykle ukládá do cookies v prohlížeči uživatele, aby bylo možné ho použít při dalším přístupu k dalším službám bez nutnosti opakovaného přihlašování.

Obrázek 4.1 Druhy různých autentizací podporovaných nativně s pomocí Visual Studio 2022



Nejdříve jsem zamýšlela použít „Individual accounts“, kde si blazor sám vytvoří lokální databázi, do které ukládá informace o uživatelích, ale tato varianta je příliš komplikované řešení tohoto problému, jelikož mimo jednoduchou správu uživatelů také nabízí dvoufaktorovou autentizaci skrze e-mail a dodatečně bych potřebovala měnit jejich kód, který interaguje s databází, jenž nebyl řádně zdokumentován.

Zvolila jsem tedy nejprve jednoduchou autentizaci, uživatel se registruje u jménem a heslem, popřípadně i s avatarem. Do databáze se uloží všechno v dané podobě a namísto hesla se v databázi uloží SHA-256 hash [3]. Přihlášení uživatele jsem řešila s pomocí JavaScriptu, kde jsem to uložila do session cookies. Toto řešení, i napříč tomu, že to nebylo cílem bakalářské práce, mělo jednu podstatnou chybu, nebyl použit jednorázový token, v browseru byl uživatel v session uložen svým uživatelským jménem. Tento fakt měl za následek to, že bylo možné si změnit tuto cookie na jiného uživatele (nebo i dokonce tam tu cookie ručně přidat) a stát se jiným uživatelem.
Takovéto řešení bylo z bezpečnostního hlediska neakceptovatelné. Tím pádem jsem musela přejít na jiné řešení, které by nebylo příliš těžkopádné, jako „Individual accounts“ nabízené Visual studiem, nebo přijít na to, jak se to dělá interně v Blazoru. Nakonec jsem se rozhodla, že použiji druhou volbu, která je z dlouhodobého hlediska ta lepší varianta.
Práce s databází a verifikace zůstává stejnou, avšak jedná se o jiné ukládání uživatelské session cookie, kde se do browseru uloží jednorázový token. Takováto verze ukládání session cookie není nejbezpečnější (nabízí se kradení cookies), ale taky se to jedná o prototypovou aplikaci, kde nemá smysl klást na důraz funkce, o které se firmy budou starat sami (například s použitím Active directory, toto je naprosto ideální pro tento účel), tím pádem toto vidím jako ideální kompromis mezi bezpečností a taktéž jednoduchostí.
A menší vsuvka, uživatelské heslo se ukládá do LocalSession, kde tato informace o přihlášení leží do té doby, než nastane smazání – buď uživatelem či aplikací. Taky se tato session cookie sdílí mezi jednot-

[3] Zde poznamenám, že SHA-256 sice na daný moment není nejsilnější hash (například existuje SHA-512), ale cílem práce, jak jsem již říkala, není neprolomitelný systém pro přihlašování (ověřování identity) uživatelů. Tím pádem SHA-256 se jeví jako dostatečně jednoduchá ochrana, která alespoň zakrývá uživatelská hesla.

livými okny prohlížeče. Naproti tomu jde použít Serverside storage, kde se tato informace uchovává jenom v jednom okně prohlížeče a při vypnutí prohlížeče se informace smažou.

Prototyp nebude podporovat změnu uživatelského hesla, jména, emailu či avataru. Toto je nadstavba, která není nutná pro řízení rizik.

Na chvíli se vrátíme k uživatelským rolím, při vytváření položek v databázi jsem se rozhodla, že uživatelé se budou dělit na „Employee“, „Admin“, „Risk management team“, „Analyst“. Struktura databáze pro uživatele a role je takováto:

Tabulka 4.1 Struktura uživatele v databázi

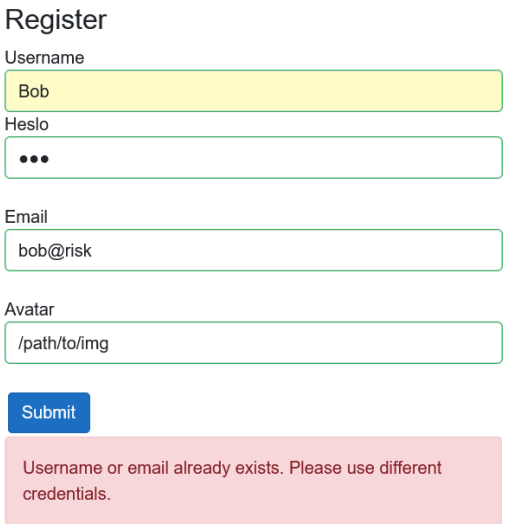
Název	Datový typ	Popis	Klíč	Null hodnoty
User_id	bigint	Identifikátor v databázi	PK	Ne
Role_id	bigint	Jaká role patří uživateli	FK	Ne
Username	nvarchar	Uživatelské jméno		Ano
Email	nvarchar	Emailová adresa		Ano
Passwd_hash	varbinary	Hash uživatelského hesla		Ano
Avatar	bigint	ID Delegáta rizika	odkaz/cesta k avataru	Ano

Tabulka 4.2 Struktura role v databázi

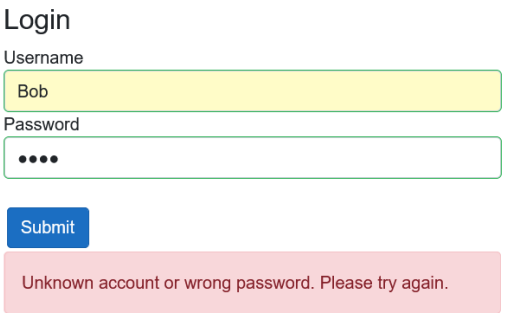
Název	Datový typ	Popis	Klíč	Null hodnoty
Role_id	bigint	Identifikátor v databázi	PK	Ne
Name	nvarchar	Název role	FK	Ne
Description	nvarchar	Popis role		Ano
Rank	bigint	Úroveň role		Ne

Po implementování registrace a přihlašování uživatele, stránky vypadají takto:

```
private readonly ProtectedLocalStorage _sessionStorage;
private ClaimsPrincipal _anonymous
   = new ClaimsPrincipal(new ClaimsIdentity());
...
public async Task UpdateState(userSession session)
{
  ...
  await _sessionStorage.SetAsync(\enquote{UserSession}, userSession);
  claimsPrincipal = new ClaimsPrincipal(
     new ClaimsIdentity(new List<Claim>
      {
      new Claim(ClaimTypes.Name, userSession.Username),
      new Claim(ClaimTypes.Role, userSession.Role)
      }
     )
  );
}
```



Obrázek 4.2 Takto stránka reaguje při již existujícím uživateli nebo již použitém emailu



Obrázek 4.3 Jak stránka reaguje na špatné přihlašovací údaje

Samotné přihlášení je řešeno takto[13]:

4.3 Rizika

Máme uživatele, už k nim můžeme přidávat rizika.

Uvedeme si hlavní údaje o riziku, co všechno chceme o riziku zaznamenávat a jaké akce s rizikem můžeme provádět:

Název Jak již bylo zmíněno v datové vrstvě, toto je potřebná část pro to, abychom měli jasnou a krátkou identifikaci pro dané riziko (mimo ID), aby uživatelé to mohli jednoduše identifikovat a také si to mohli zapamatovat (narozdíl od ID).

Popis Více specifický popis narozdíl od názvu.

Garant (Delegát) V literatuře se dá setkat i s pojmem „risk owner“. V předchozích kapitolách byla vysvětlena role garanta a delegáta

Datum další kontroly Toto políčko se používá pro případ, kdy je riziko již „Resolved“ (vyřešené). Když nastane datum další kontroly, pak se toto rniziko znovu nastaví jako nové riziko.

Pravděpodobnost a dopad Normální škála zde bude 1 až 5. Uživatelé si potom budou moci sami nastavit škálování.

Jiné bodování Respektive jiný druh skóre. Bude možné si donastavit jiný druh bodování v případě, že pravděpodobnost/dopad nejsou dostatečným druhem hodnocení.

Status U rizika nás také zajímá, v jakém stavu je.

Původ Je zde také nutné uvést původ rizika, například penetrační testování, či ostatní typy nálezů.

Přiložené soubory Taktéž je vhodné mít možnost přidat reporty a podobné soubory k riziku.

Tags Které tags (štítky) bude mít riziko, aby rizika byla víc jednoznačná a snadněji rozlišitelná.

Protiopatření Bude to list jednotlivých protiopatření

Historie Je nutné si zaznamenávat historii rizika. Zajímá nás kdo, kdy a co provedl.

Update Update rizika znamená, že s tím rizikem nějak makládáme, posouváme riziko dál nebo mu dáváme další vlastnosti (například tagy).

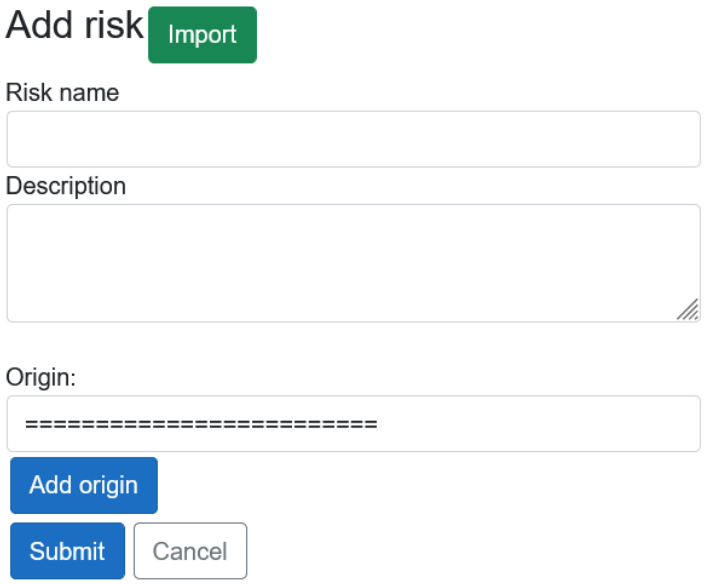
Edit Změnou rizika se myslí změna názvu, který například nebyl dostatečně výstižný, popisu nebo doplnění původu.

Delete Můžeme i smazat riziko.

Takto jsem navrhla entitu pro reprezentaci rizika.

Nejdříve jsem začala formulářem na vytváření rizik:

Obrázek 4.4 Formulář na založení/změnu rizika



Tabulka 4.3 Struktura rizika v databázi

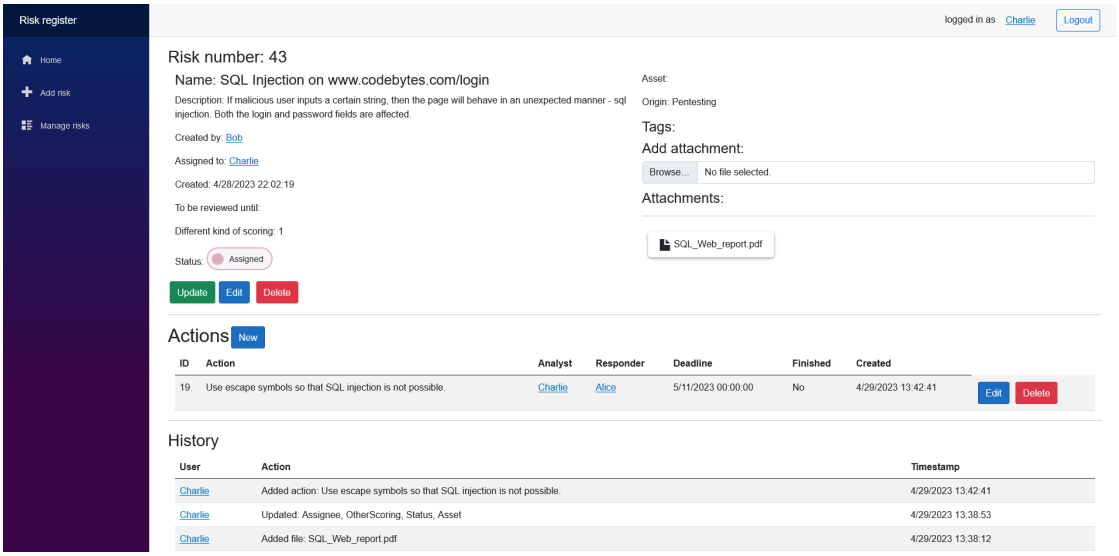
Název	Datový typ	Popis	Klíč	Null hodnoty
Risk_id	bigint	Identifikátor v databázi	PK	Ne
Name	nvarchar	Název rizika		Ne
Description	nvarchar	Popis rizika, ve formátu „Když X, pak Y.“. Popřípadně i další info		Ano
CreatedBy	bigint	ID uživatele (resp. i systému), který založil dané riziko	FK	Ne
Assignee	bigint	ID garanta rizika	FK	Ano
Delegate	bigint	ID Delegáta rizika	FK	Ano
ReviewUntil	datetime	Do kdy se musí vytvořit nějaké rozhodnutí		Ano
Created	datetime	Den a čas vytvoření rizika		Ne
Impact	int	Číslo vyjadřující dopad. Škála závisí na daném nastavení aplikace		Ano
Probability	int	Číslo vyjadřující pravděpodobnost toho, že se riziko naplní		Ano
OtherScoring	nvarchar	Vlastní způsob poznamenání závažnosti rizika		Ano
Status_id	bigint	V jakém stavu se riziko nyní nachází	FK	Ne
Asset_id	bigint	Riziko se váže na nějaké aktivum	FK	Ano
Origin	bigint	Původ rizika		Ne

V případě, že nastane chyba, aplikace zobrazí chybové hlášení, v opačném případě se otevře stránka s novým rizikem. Stejný formulář se používá i pro změnu (Edit) rizika, s tím, že má jiný název a nezakládá nové riziko.

Co se týče interní reprezentace v aplikaci, měla jsem na výběr 2 možnosti, klasický přístup typu ADO.NET oproti Entity framework. Rozdíl je v tom, že u ADO.NET píšeme ručně SQL příkazy a pracujeme s daty přímo, sami si je provazujeme a určujeme, jak s tím budeme zacházet. Entity framework je nadvrstva k ADO.NET, ale už zde nepoužíváme klasické SQL dotazy, dotazujeme se pomocí C#, je tedy více abstraktní.
Vybrala jsem si ADO.NET přístup, jelikož byl pro mě jednodušší na implementaci a mám s tím dobré zkušenosti. Je možné v budoucnu přejít na jiný přístup s pomocí Entity frameworku, ale prozatím všechny třídy, které reprezentují entity v databázi, používají CRUD [4].

Po úspěšném založení nebo změně rizika se vrátíme na stránku pro riziko.

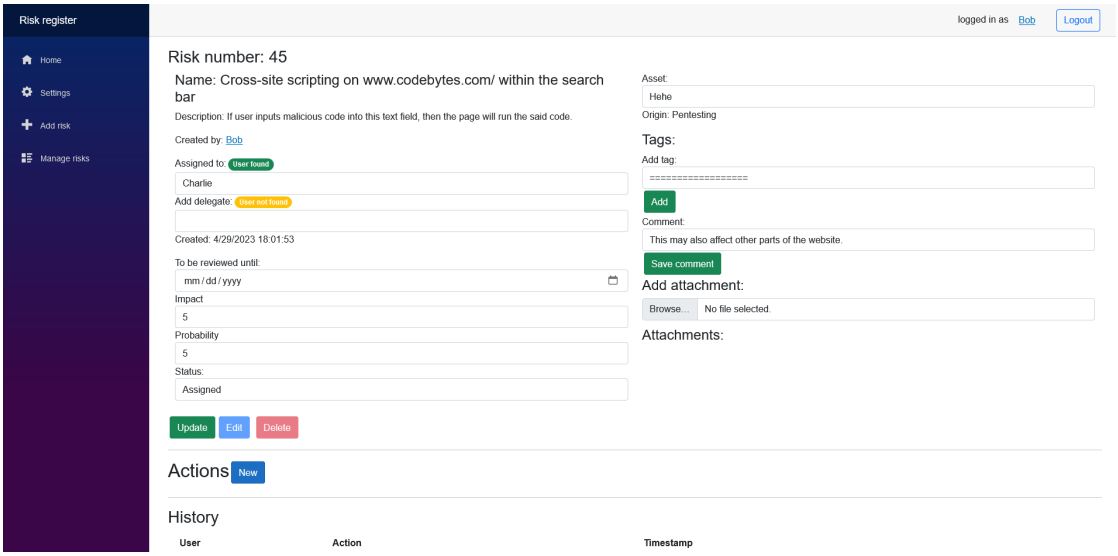
Obrázek 4.5 Finální podoba stránky pro riziko



Nyní jsem se rozhodla zavést tzv. „update“ režim stránky pro riziko, kde lze v této stránce editovat některé položky, jako „Assigned to“ (garant), popřípadně delegát, Asset (aktivum), „Review until“ (datum příští kontroly) a status. Mezi tímto update lze i zanechat komentář. // Jak „edit“ tak i „update“ akce se logují, mimo jiné.

[4] Create, Read, Update, Delete – popis základních funkcí programu

Obrázek 4.6 Podoba stránky pro „Update“



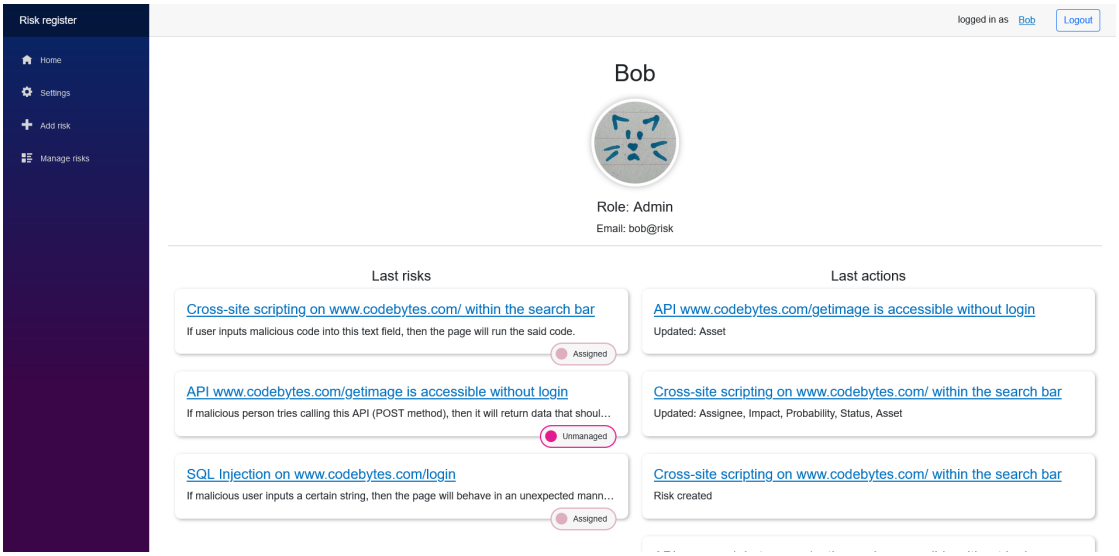
Rychle se dotknu Logů, jedná se o strukturu zatím použitou jenom pro rizika, vypadá následovně:

Tabulka 4.4 Struktura role v databázi

Název	Datový typ	Popis	Klíč	Null hodnoty
Log_id	bigint	Identifikátor v databázi	PK	Ne
Action[5]	nvarchar	Text popisující událost		Ne
User_id	bigint	Uživatel, který provedl akci	FK	Ne
Risk_id	bigint	Kterého rizika se to týkalo	FK	Ne

Logy lze vidět v historii rizika nebo u uživatelů na jejich stránce jako například zde:

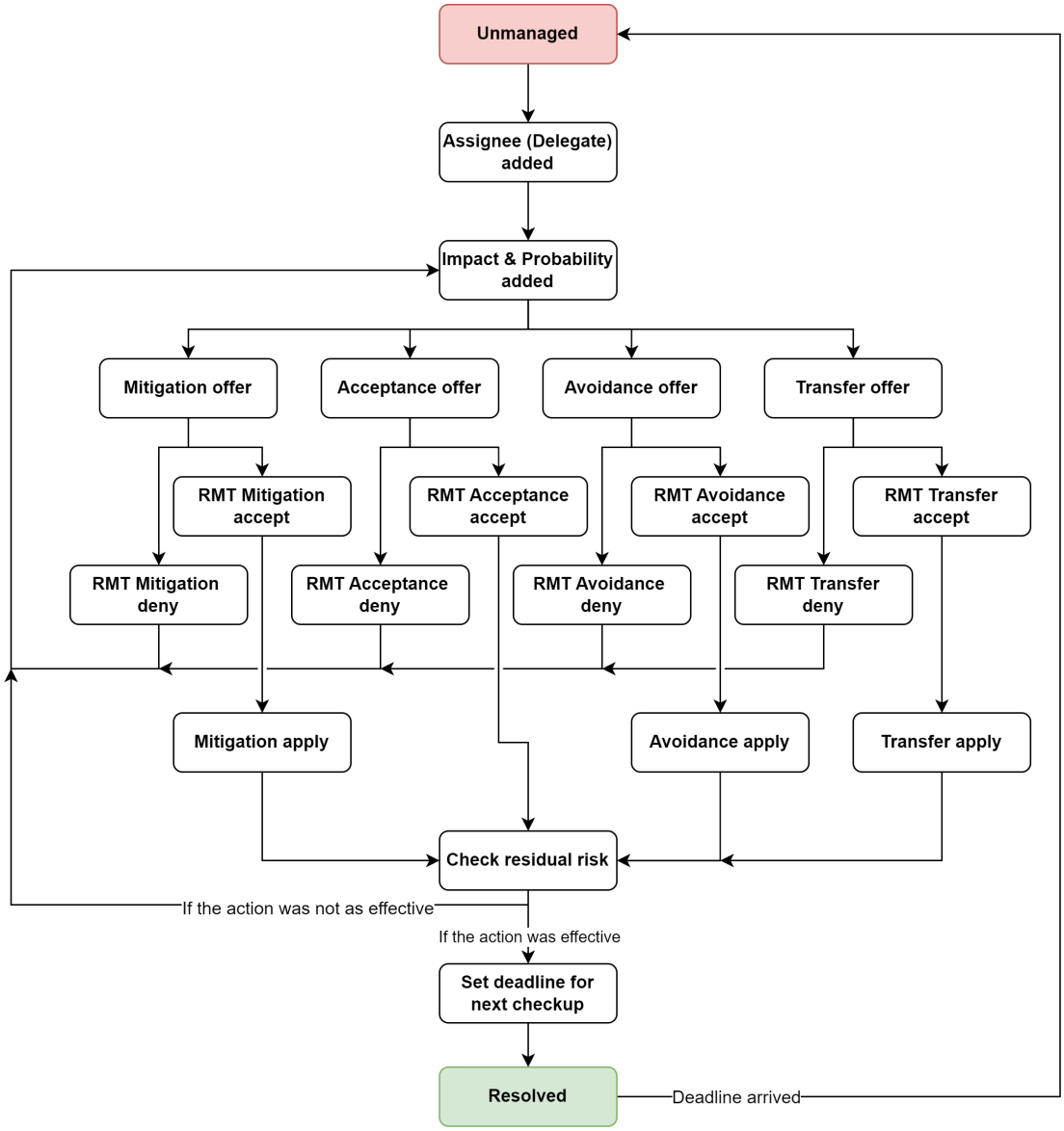
Obrázek 4.7 Uživatel s událostmi



Při „Update“ lze vidět jednotlivé položky, které se změnily a v některých případech jak se změnily. V případě, že uživatel zanechá komentář, takový komentář se ukládá jako Log a bude vidět v historii rizika. V rámci toho, že to je log, je nemožné bez přístupu do databáze změnit tyto údaje nebo je smazat.

Po implementaci „Update“ režimu jsem řešila stav rizika. Vzhledem k analýze procesu rizik jsem navrhla tyto jednotlivé stavy. U těchto rizik je nutné vědět, kdo může měnit jednotlivé stavy. Kvůli vysoké náročnosti této featury jsem ji zanechala jenom jako concept, ale v reálném podniku některé stavy by mohli nastavit jenom uživatelé z jistých skupin (jako například stav „RMT Transfer accept“ může nastavit jenom RMT). Plus je možné automatizovat stavy a přidat různá tlačítka například pro RMT, kde by mohli potvrdit nebo okomentovat riziko. Ale stejně jako předtím, časově takovéto featury nejsou možné v rámci bakalářské práce. O dalších konceptech, které by se hodily do finální podoby aplikace viz sekce Další možné featury. Kontrola změny stavů je ponechána na Logu.

Obrázek 4.8 Zobrazení možných stavů pro dané riziko



Aplikace nabízí také různé typy hodnocení rizik a pokud je vybrané normální hodnocení, takže v případě, že v nastavení aplikace je nastavené normální hodnocení, pak zde budou 2 textboxy - „Probability“ a „Impact“. V případě, že uživatel zadal příliš velké nebo malé číslo (to jest menší jak 1 nebo větší jak 5) se zobrazí div varující uživatele o tom, že tam zadal špatné číslo. Podobné se dějě i u jiného druhu hodnocení, kde je na uživateli, aby vypočítal hodnotu rizika, a zadal číslo. Později se k tomu dostaneme blíže.

Některé položky slouží jako „placeholder“. Ještě k nim chybí propojení na stránky, na kterých je budeme moci měnit (tagy, aktiva, protiopatření, nahrávání souborů a i stavy).

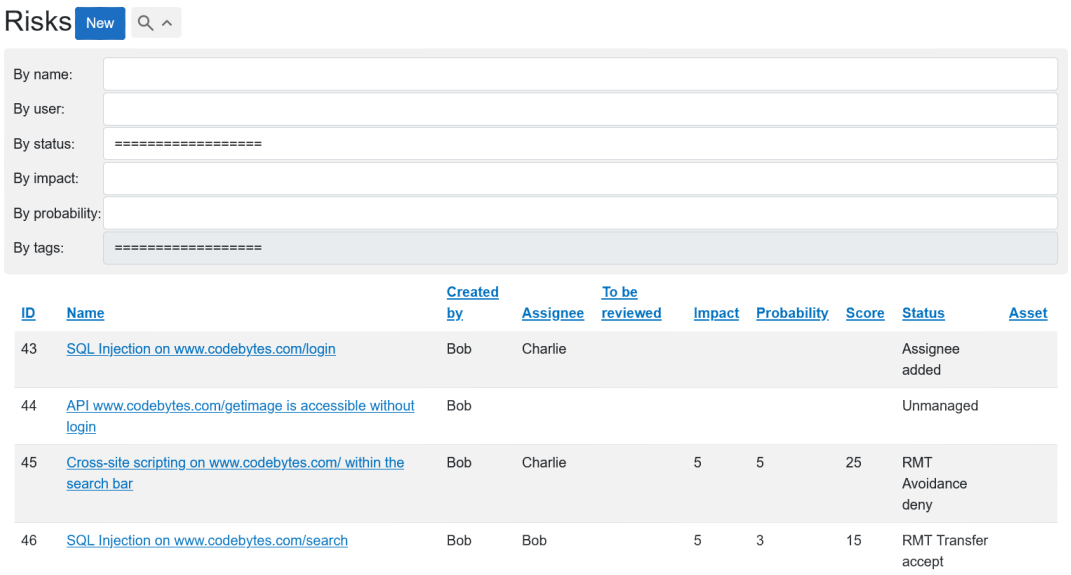
Při testování jsem vytvořila menší počet stavů, které bylo možné editovat jenom v SQL nástroji.

4.3.1 Seznam rizik

Jako další část jsem navrhla a naprogramovala stránku, která zobrazuje rizika a popřípadně je i filtruje.

Hledání zde nefunguje na přesnou shodu, jenom na počáteční řetězec (v C# funkce „StartsWith“).

Obrázek 4.9 Seznam rizik



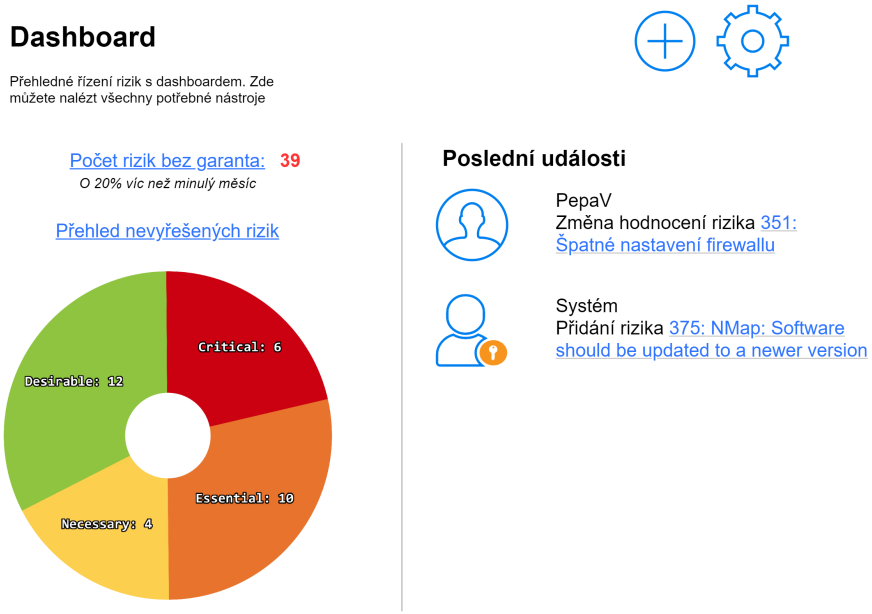
Měla jsem v plánu ještě doimplementovat hledání dle tagů, ale takovéto hledání je náročné na implementaci, tím pádem jsem to nechala jako další návrh do Další možné featury.

Jak to je vidět ve snímku, tak lze vyhledávat dle názvu rizika, dle uživatele (bude vyledávat ve sloupcích „Created by“ a „Assignee“). Toto vyhledávání je také case sensitive. Dále dle stavu rizika a Dopad/Pravděpodobnost. Hledání funguje na bázi logického „AND“, tím pádem se zobrazí rizika, která splňují všechny předpoklady.

4.3.2 Dashboard

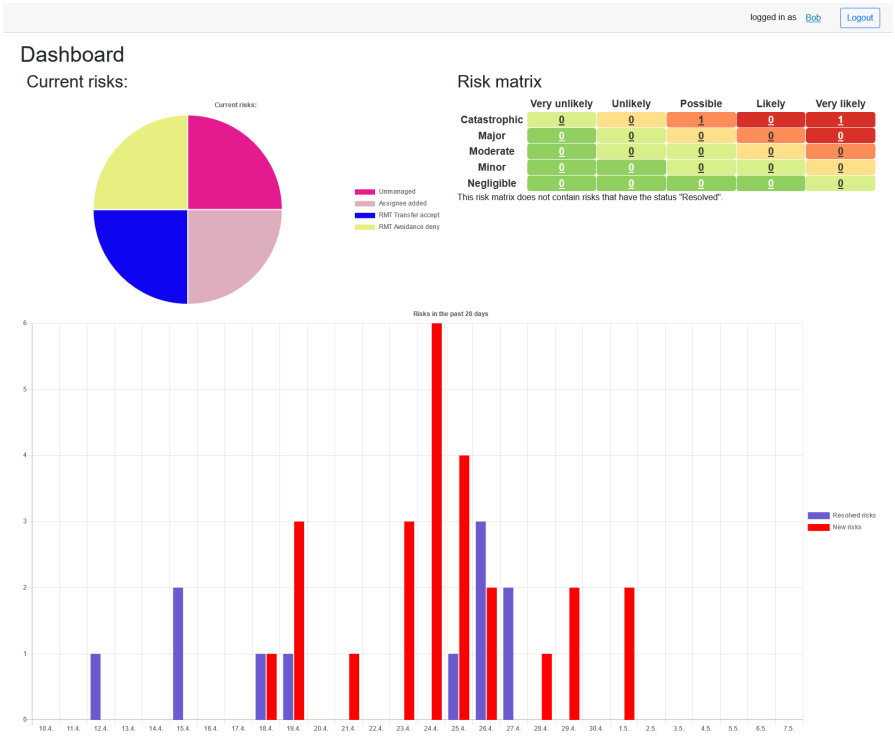
Dashboard by měl být informativní, přehledný a měl by obsahovat informace, které jsou nejvíc kritické. Zde je vidět první návrh Dashboardu, vytvořený ještě před návrhem procesů pro řízení rizik:

Obrázek 4.10 První návrh designu dashboardu



Jelikož jsem při analýze řízení rizik našla Risk Matrix, tak jsem se rozhodla, že nahradí část s „Poslední události“. Ale levou část s přehledem nevyřešených rizik jsem naprogramovala do aplikace. Samotný Dashboard ve finále vypadá takto:

Obrázek 4.11 První návrh designu dashboardu



V seznamu nynějších rizik jsou vidět rizika, která nemají stav „Resolved“ (stejně tak jako v matici).

```
<AuthorizeView>
    <NotAuthorized>
        This part is visible to an anonymous user
    </NotAuthorized>
    <Authorized>
        This part is visible to a logged in user
    </Authorized>
</AuthorizeView>
```

```
@attribute [Authorize]
```

Při kliknutí na nějakou položku z Risk matrix se uživatel dostane na seznam rizik s již nastaveným filtrem. Toto je udělané s pomocí parametrů ve stránce.

Pod těmito dvěma prvky se nachází graf, zobrazující počet nových a vyřešených rizik za posledních 28 dnů.

Koláčový graf a sloupcový graf byly vytvořené s pomocí ChartJS, balíčku pro razor, jelikož javascript není jednoduché zprovoznit v Blazoru, obešla jsem to s tímto způsobem. ChartJS je publikovaný pod MIT licenzí, takže jej můžu používat v této aplikaci.

V případě, pokud skóre rizik je nastavené na jiné, než normální, pak Risk matrix nebude dostupná a místo risk matrix se zobrazí tento prvek:

Obrázek 4.12 První návrh designu dashboardu

> Risk matrix is only supported with the normal type of scoring.
> In case you want to display your kind of matrix scoring, please implement your own solution.

4.4 Omezení stránek a nastavení

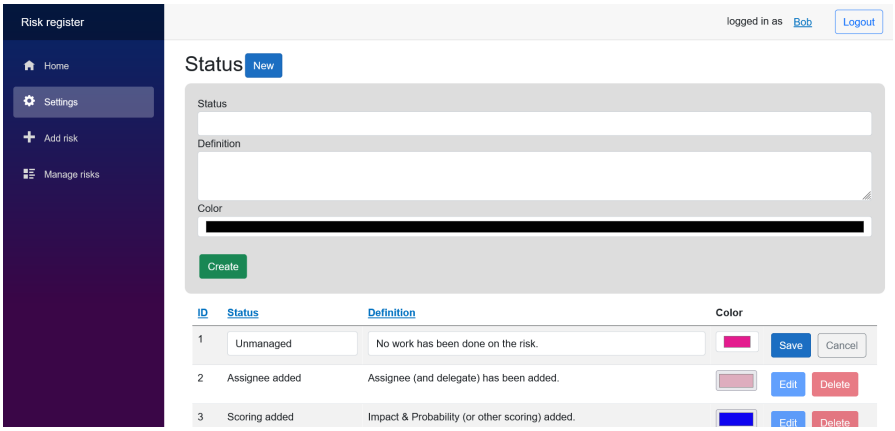
V této části si projdeme, jak jsem v Blazoru vyřešila problém s řízením přístupů a další stránky aplikace, které by měly být přístupné pod zvýšenými právy.

Nebo alternativně přidat tento tag na začátek souboru: V druhém případě, pokud uživatel není přihlášený, tak se mu zobrazí prázdná stránka.
Dále, pokud chceme, aby stránka byla přístupná jenom Adminstrátorovi, lze to vyřešit takto: Mezi další změny, které zde přibily díky rolím, patří i přidání nové položky do lišty, kde se dá nyní najít „Settings“ pokud má uživatel roli „Admin“. Nepřihlášený uživatel nemá žádné role.

Jedna z částí, která spadá pod navýšená práva, je spravování stavů rizik. Tato stránka slouží pro zakládání nového stavu a i pro editaci či mazání. Při mazání jsem implementovala tu funkcionalitu, že stavy, které jsou použité u alespoň jednoho rizika nelze smazat.

```
@attribute [Authorize(Roles = \enquote{Admin})]
```

Obrázek 4.13 Stránka pro editaci stavů rizik



Barvy stavům jsem přidala z toho důvodu, aby v případě potřeby byly visuálně rychlejší k rozeznání a aby bylo možné reagovat rychleji na nějaké druhy stavů.

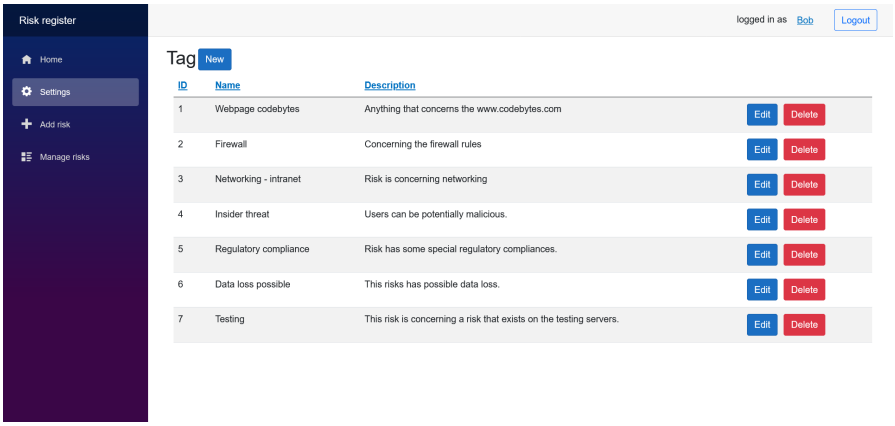
Změny v těchto částech nevytváří další záznamy do logů, jelikož se nejedná o riziko, ale hodilo by se to do oddělené tabulky v databázi. Zde je databázová struktura k stavům:

Tabulka 4.5 Struktura role v databázi

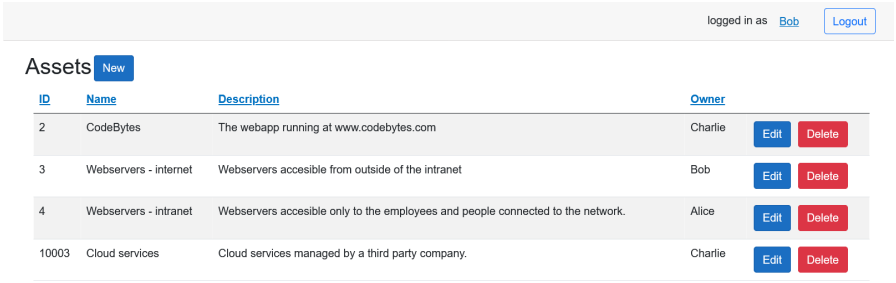
Název	Datový typ	Popis	Klíč	Null hodnoty
Status_id	bigint	Identifikátor v databázi	PK	Ne
Status	nvarchar	Název stavu		Ne
Definition	nvarchar	Popis stavu		Ano
Color	nvarchar	Barva v hex formátu		Ano

Dále zde jsou tagy a aktiva, ale to jsou jednoduché CRUD stránky se základní logikou.

Obrázek 4.14 Seznam tagů



Obrázek 4.15 Seznam aktiv



Menší poznámka k indexu, toto je známý bug v indexaci v Microsoft SQL serverech.

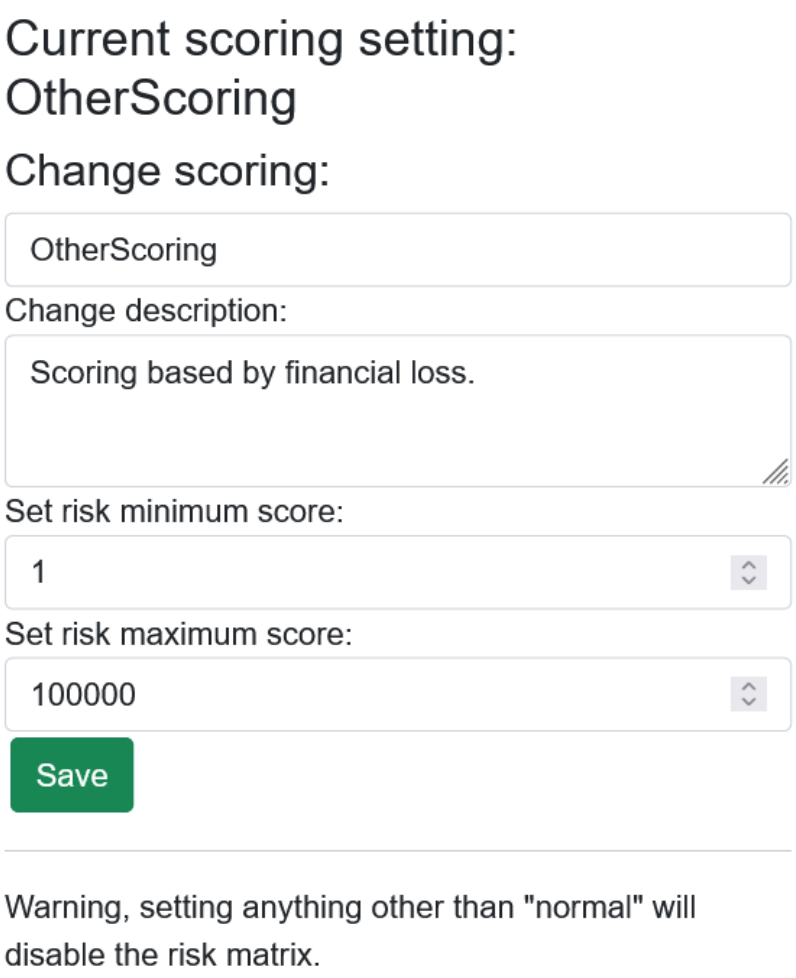
4.4.1 Změna bodování

V anglickém překladu „Change scoring“. Jedná se o změnu bodování rizik. Pro ukládání změn rizik se narozdíl od ostatních dat používá .json soubor, jehož struktura je takováto:

```
{
  "Scoring": {
    "Current": "OtherScoring",
    "Description": "Scoring based by financial loss.",
    "Min": 1,
    "Max": 100000
  },
  "PossibleScoring": {
    "Normal": "Normal scoring based on impact and probability.",
    "CWSS": "Common wea kness scoring system",
    "CVSS": "Common vulnerability scoring system",
    "OtherScoring": "Scoring based on your own criteria,    change    this description so that
  }
}
```

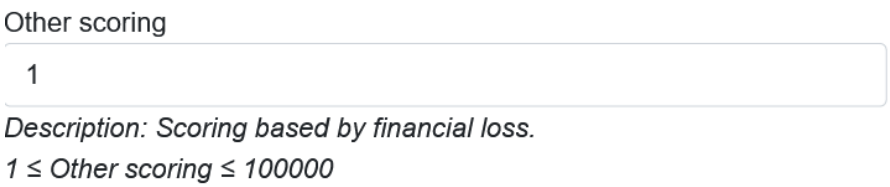
Zde jsou vidět tyto 4 možnosti, které si administrátor může vybrat. U normálního nelze nastavit minimální a maximální hodnoty, ale u ostatních to je možné. Totéž ale neplatí i pro popis bodování, v případě jiného bodování (není myšlen doslovný překlad „OtherScoring“) lze změnit popis v libovolném místě.

Obrázek 4.16 Změna bodování – ukázka



Tato změna má vliv na několik částí aplikace, na hlavní stránce se buď zobrazí Risk matrix nebo varovný prvek, v seznamu rizik se zobrazí buď jeden sloupec v případě, že není vybrané normální bodování nebo naopak klasické „Impact“ a „Probability“. Poslední změna se „Update“ rizika (a taky mimo update režim), kde se pod textovým políčkem s hodnocením zobrazí popisek, který jsme tam vložili:

Obrázek 4.17 Změna bodování - ukázka

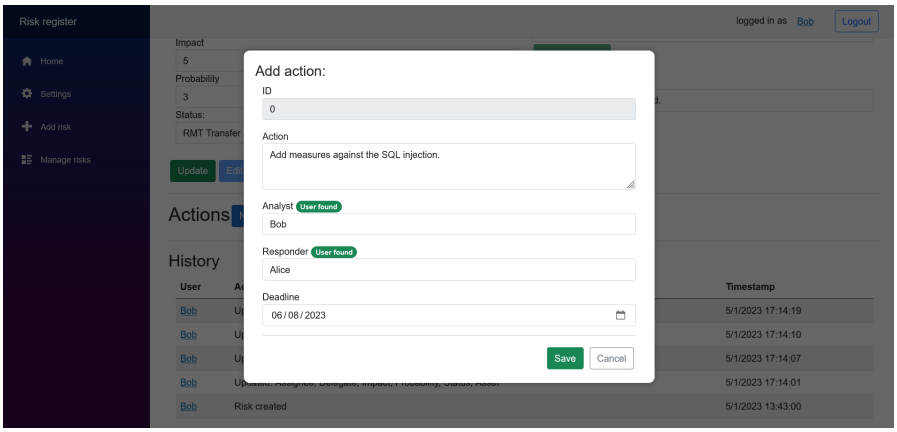


4.5 Finalizace

Z pohledu dokumentace se již blížíme ke konci, tím pádem se vrátíme k rizikům a doplníme pár funkcionalit, o kterých jsme si předtím něco jenom říkali.

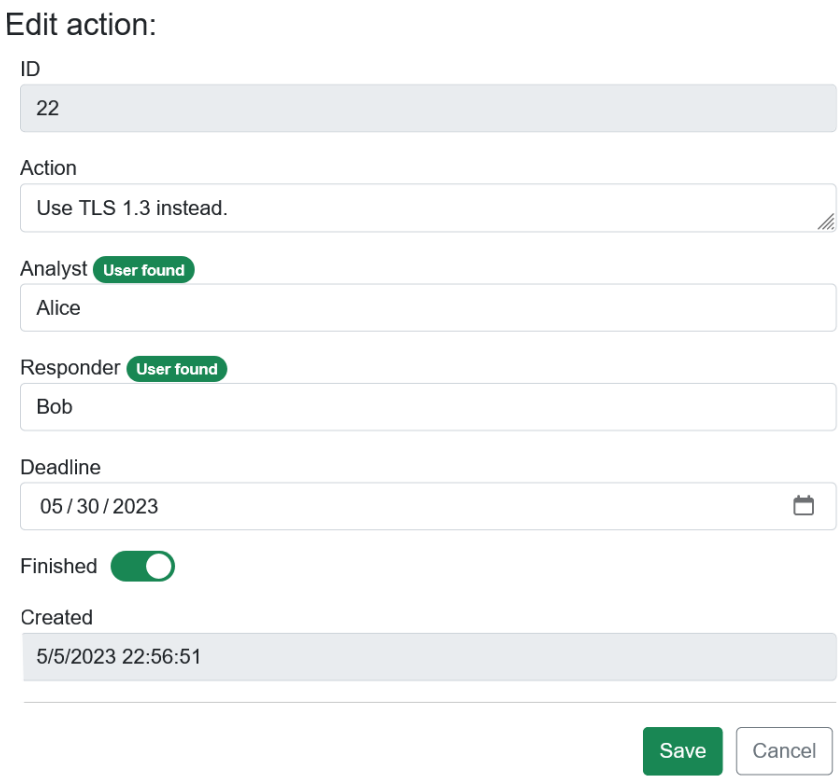
První funkcionalitou jsou „Actions“ neboli protiopatření. Zde uživatel může zadat nové protiopatření, ve kterém figurují dvě osoby. První je „Analyst“ (analytik) neboli člověk, který dané protiopatření navrhnul (na základě vlastních znalostí či pozorování). Naproti tomu je „Responder“ (responder) je osoba, která odpovídá za implementaci daného protiopatření. Doplní se popis daného protiopatření a stanoví se datum, do kdy dané protiopatření se musí splnit. K jednodtlivému riziku lze přiřadit vícero protiopatření s různými uživateli, aby bylo možné lépe určit co je čí zodpovědnost a nemusela pak kvůli tomu probíhat ve firmě další komunikace.

Obrázek 4.18 Změna bodování – ukázka



Při případné změně protiopatření se zobrazí další políčko, kde se dá zakliknout, zda dané protiopatření bylo splněno.

Obrázek 4.19 Změna bodování – ukázka



Protiopatření jsou pak uložená v databázi jako další entita:

Tabulka 4.6 Struktura role v databázi

Název	Datový typ	Popis	Klíč	Null hodnoty
Action_id	bigint	Identifikátor v databázi	PK	Ne
Action	nvarchar	Popis protiopatření		Ano
Analyst	bigint	ID uživatele, který navrhnul protiopatření	FK	Ano
Responder	bigint	ID uživatele, který zodpovídá za splnění	FK	Ano
Deadline	datetime	Datum splnění akce		Ano
Finished	bit	Indikátor dokončení akce		Ne
Created	datetime	Čas vytvoření opatření		Ne
Risk_id	bigint	Kterého rizika se to týkalo	FK	Ano

Jako další část, která nám zde zbývá, je nahrávání souboru a stahování souboru. Jelikož z Blazoru nejde jednoduše volat Javascript funkce (blazor se funguje dost specificky v tomto ohledu), se to muselo trošku obejít. Při nahrávání souboru nemusíme používat Javascript, pouze posloucháme event:

```
private async Task LoadFiles(InputFileChangeEventArgs e) {
  loadedFiles.Clear();
  foreach (var file in e.GetMultipleFiles(1)) {
    loadedFiles.Add(file);
    if (file.Size > 0) {
      using var stream = new MemoryStream();
      using var reader = new StreamReader(file.OpenReadStream());
      await reader.BaseStream.CopyToAsync(stream);
      var contents = stream.ToArray();
      UploadContents(contents);
    }
  }
}
```

Jedná se sice o zjednodušenou variantu kódu, než který používám, ale ukazuje, jak přibližně lze nahrát soubor například do databáze.

Ohledně stahování souborů, potřebujeme javascript kód a pak tento javascript volat:

```
function downloadFile(fileName, data) {
const link = document.createElement('a');
link.download = fileName;
link.href = 'data:application/octet-stream;base64,' + data;
document.body.appendChild(link);
link.click();
document.body.removeChild(link);
}
```

a pak C# kód, který to stahuje:

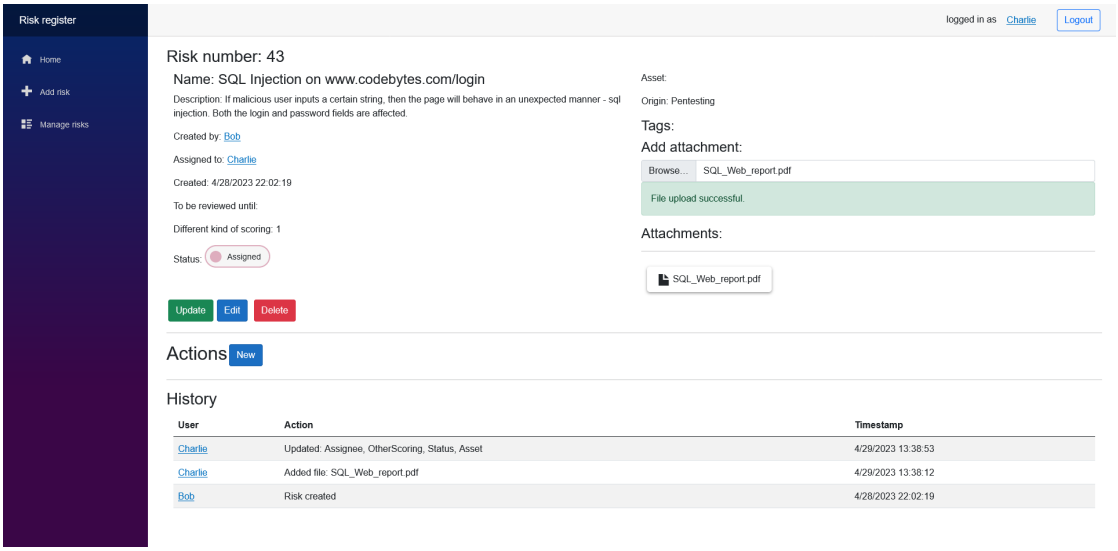
```
async Task FileDownload(FileC file)
{
  var export = Convert.ToBase64String(file.Contents);
  await JS.InvokeVoidAsync("downloadFile", file.Filename, export);
}
```

Zde je použita FileC třída, která obsahuje objektovou reprezentaci entity v databázi. Rozhodla jsem se, že prvky budou uložené v samotné databázi:

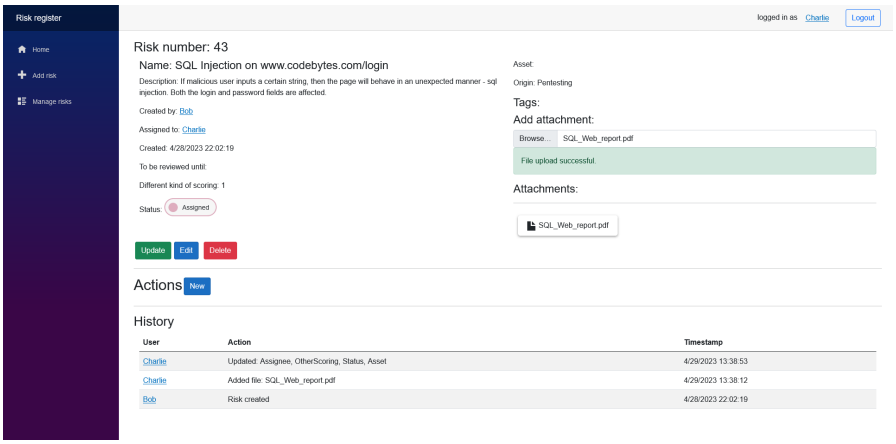
Tabulka 4.7 Struktura File v databázi

Název	Datový typ	Popis	Klíč	Null hodnoty
File_id	bigint	Identifikátor v databázi	PK	Ne
Filename	nvarchar	Název souboru		Ne
Contents	varbinary	Obsah souboru		Ne
Risk_id	bigint	Ke kterému riziku patří tento soubor	FK	Ne
User_id	bigint	Uživatel, který přidal soubor		Ano

Obrázek 4.21 Mazání souboru je možné.



Obrázek 4.20 Jak se stránka chová při nahrání souboru.



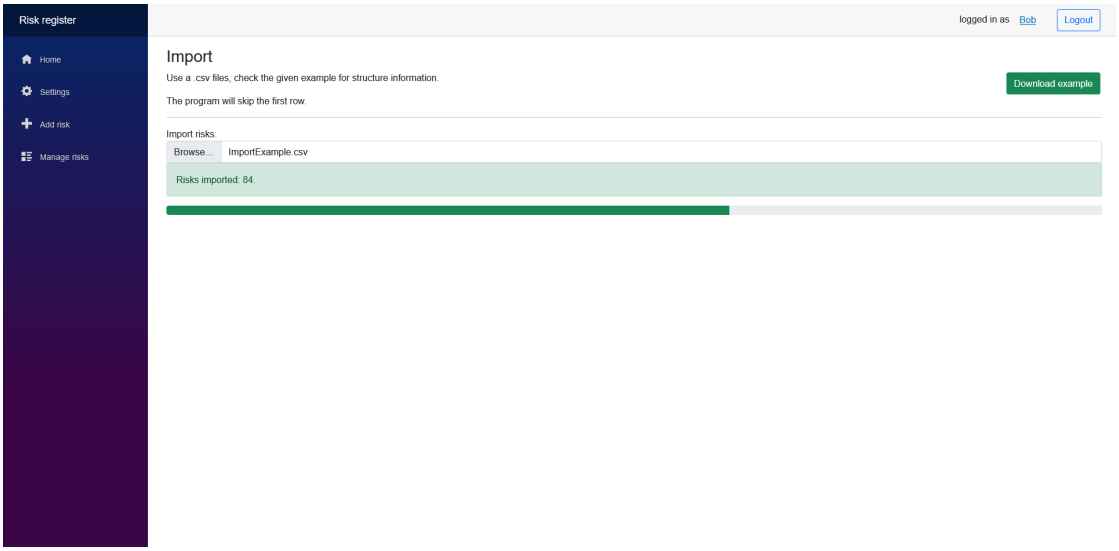
Na on-hover události se zobrazí i křížek, s pomocí kterého se dá tento soubor smazat a zobrazí se nápověda, který uživatel nahrál daný soubor.

Tento systém má jednu vadu – v případě, že z jednoho souboru pochází víc rizik, tak dochází k duplikaci informací. Toto si bude muset rozmyslet vývojář či analytik, jak to mají domluvené v podniku, jaká je kardinalita mezi rizikem a a souborem. Dle toho se pak může rozhodnout a řešení by bylo podobné jako tagy.

Poslední feature, kterou jsem implementovala, byl formulář na import:

```
builder.Services.Configure
  <Microsoft.AspNetCore.Components.Server.
    CircuitOptions>(options => {
      options.MaxBufferedUnacknowledgedRenderBatches = 500;
    }
);
```

Obrázek 4.22 Mazání souboru je možné.



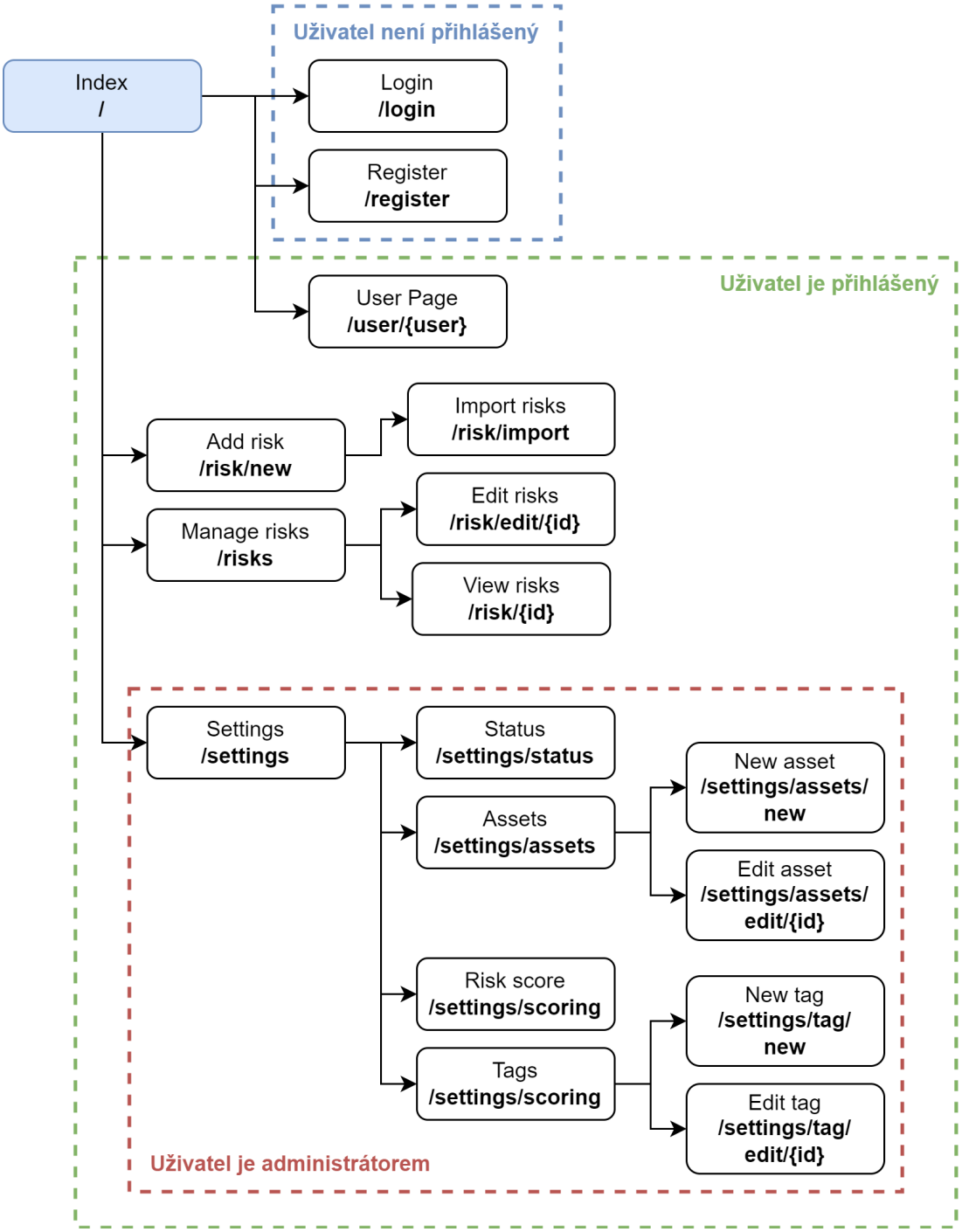
Kód, který obstarává stahování a nahrávání souborů, byl použit ten, který sloužil u rizik jako „Attachments“. „Download example“ tlačítko stáhne .csv soubor, který je uložen jako string za běhu programu. Program se v jistém bodě bude chovat ne podle očekávání – jestli má uživatel více jak 500 rizik v jednom souboru, stránka se přestane obnovovat a obnoví se, až se všechny rizika donačtou. Toto je způsobené tím, jak Blazor interně funguje. Lze to ovlivnit, počáteční hodnota je 10 obnovení.

Takto jsem tuto hodnotu navýšila na 500, což je dle mého názoru v pořádku limit[14].

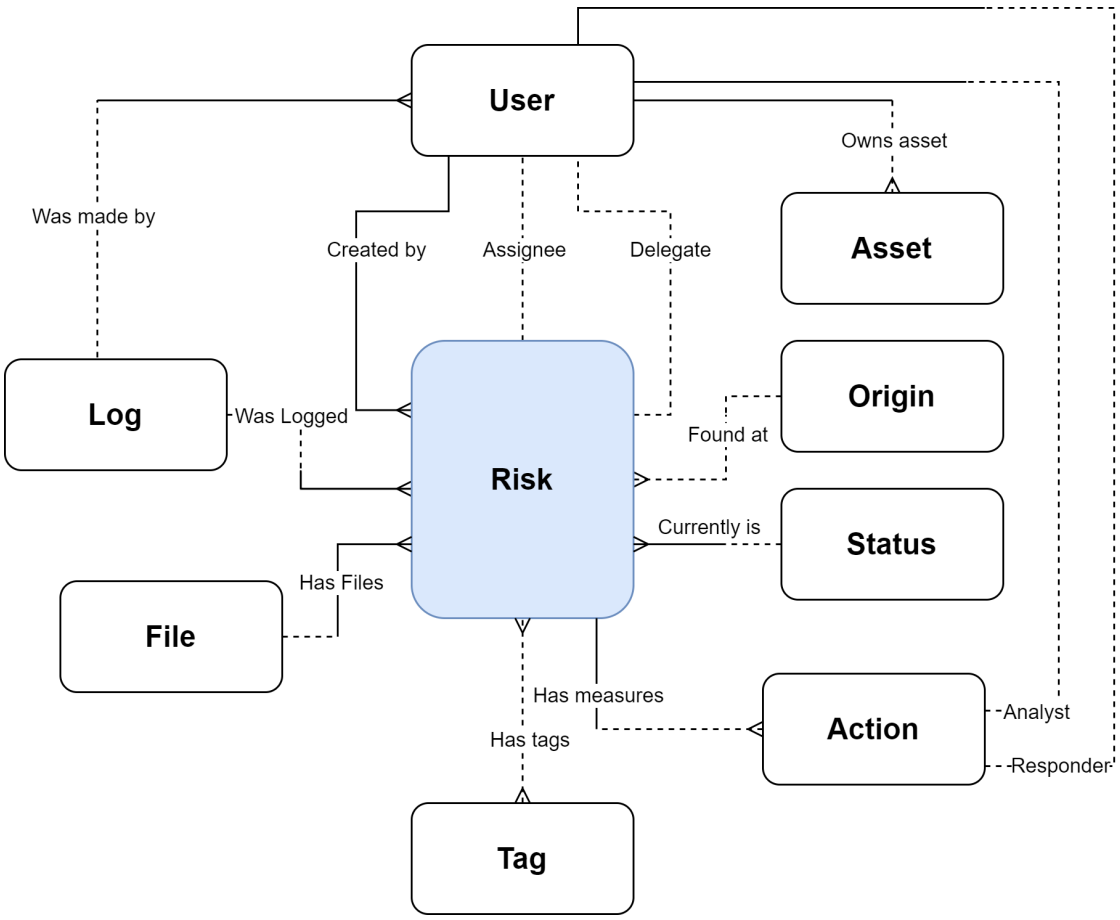
4.6 Závěr analýzy aplikace

Ve finále zde je vidět mapa stránek:

Obrázek 4.23 Mapa stránek



Obrázek 4.24 Zjednodušený model datové vrstvy, reprezentující jednotlivá propojení mezi entitami



Zde je znázorněna kardinalita a propojení jednotlivých entit.

Některé entity jsem zde nevypisovala, protože nebyly dostatečně podstatné.

V momentě, kdy jsem se rozhodovala mezi ADO.NET a Entity frameworkem, tak jsem se rozhodla špatně, měla jsem si vybrat Entity framework. Nejvíce stráveného času bylo na ladění jednotlivých dotazů a tento čas mohl být využit efektivněji.

4.7 Další možné featury

Zde mám nápady, které buď byly časově příliš náročné na implementaci, nebo které se objevily až v pozdější fázi implementace. Implementace všech těchto featur by zabrala velké množství času a na to není čas.

Víc stránek o uživatelích Jelikož moje implementace uživatelů je velmi bare-bones a nějaký list uživatelů, víc akcí by bylo velmi potřebné v reálné aplikaci. Stránka, která by byla přístupná jenom přihlášenému uživateli na url (například) /dashboard by se uživateli zobrazil vlastní dashboard, kde by viděl vlastní rizika, jejich progres a hlavně také, na kterých protiopatřeních má pracovat nebo je zavést.

Export rizika Zdánlivě se jedná o jednoduchou úlohu, ale jenom zdánlivě. Ideálně by export rizika byl konfigurovatelný, od výběru jednotlivých položek až po výběr jednotlivých formátů, avšak toto už je časově velmi náročné.

Spravování rizik ze seznamu rizik Tato feature by byla o tom, že v seznamu by byl checkbox, s tím, že všechna vybraná rizika šlo smazat, nebo nastavit „Review until“ apod.

Automatická změna stavu Této featury jsem se již dotkla v dřívější části, nejedná se o stavy jako „Assignee (Delegate) added“ (který by mohl být změněn), ale například když přijde deadline „ReviewUntil“ tak se riziko resetuje na „Unmanaged“ z „Resolved“. Toto by se mělo dělat skripty, a nejsem si jistá, jestli toho je možné docílit v webové aplikaci jako Blazor. Ale jednoduchý skript volající SQL query by stačil.

Lepší integrace rolí Během vývoje jsem využila jenom velmi hrubé dělení Uživatel-Admin, 2 zbylé role zůstaly nevyužity. Jako novou featuru bych navrhla, že stav a role dostanou n:m tabulku, kde jednotlivé role budou moct přiřazovat stavy jenom dle svých rolí a podle nějakého schématu. Či naopak je také možné propojit stav s tlačítky, například v situaci, kdy jsme členem RMT a můžeme odmítnout či schválit mitigaci, tak dostaneme dva tlačítka – „Accept“/„Deny“. Pak by se z této části stal automat. Takový automat už by měl znatelně jinou strukturu než ten daný mnou a raději přenechám takovou případnou strukturu čtenáři.

Emailování událostí Další možnou featurou se jedná posílání zpráv na email uživatele. Když se uživatel stane „Responder“ u protiopatření, dostane email a podobně. Při možné implementaci je zde nutné si dávat pozor, aby počet emailu byl logický, jinak uživatel bude ignorovat emaily, možná je i odhazovat do SPAM složky, což je kontraproduktivní, jestli chceme mít efektivní řízení rizik ve firmě. Emailování není jediný způsob kontaktování, pokud se ve firmě používají jiné služby, tak lze využít i ty.

Napojení na risk analysis programy/služby S touto feature je nutno nakládat opatrně, protože jestli se implementuje, je velmi důležité, aby se nevytvářely duplikáty v risk registru.

Kapitola 5
Testování

Alice: „ Už máme provedenou analýzu rizik a funkční aplikaci.“
Bob: „Tak to můžeme jít na testování!“

Posledním cílem této bakalářské práce je otestovat prototyp v reálném prostředí. Prototyp je navržen obecně pro menší podniky a s tím přichází i menší problémy. Pro podnik je možné se pokusit odhadnout přibližnou formu či strukturu informací o rizicích, avšak není možné docílit toho, aby byl jeden nástroj pro všechny. Je tedy nutné tyto změny obejít či implementovat, respektive s nimi naložit.

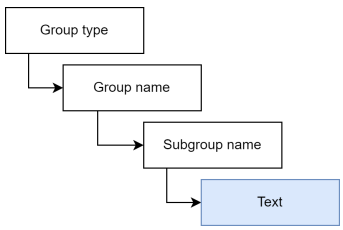
Taktéž jsem vytvořila protypovou aplikaci, kde v některých místech má tato aplikace jenom základní funkcionalitu oproti komerčně dostupnému software.

5.1 Testování na skutečných datech

5.1.1 Reprezentace dat

K testování jsem použila data uložená v .xlsx souboru exportovaných z nástroje RAMSES[15]. RAMSES je ISMS nástroj primárně kna analýze rizik. Jelikož se nejedná o efektivní nástroj pro správu rizik (nástroj to podporuje, ale není nejlepším řešením), tak se tyto rizika exportují do aplikace pro řízení rizik, v tomto případě náš prototyp.
Jak již jsem nastínila na začátku kapitoly, budeme se potýkat s různou reprezentací dat. Toto je zjednodušená reprezentace záznamu, který pochází z RAMSESu:

Obrázek 5.1 Kategorizace rizika v podniku



Kde políčko „text“ reprezentuje popis rizika a nadřazená políčka jsou jiné entity. Organizace je ID a název organizace, Group type obsahuje taky ID a název, Group name je jenom název – bez ID, Subgroup obsahuje ID, název a popis.

Dále záznam patří do kategorie rizik. Podobně jako naše aplikace, záznam je v nějakém stavu (reprezentováno číselníkem vedoucím na jinou entitu). Je taky Provázaná k aktivu přes vlastní asset_id a

název aktiva. Záznam má taky garanta (u nás se to řeší přes garanta a delegáta) a čas poslední změny (u nás je taky teoreticky čas poslední změny, ale ten je vidět v historii rizika). Zde pak přichází hlavní rozdíl mezi prototypem a RAMSES záznamem - každé „riziko“ (respektive záznam) má položku „comment_txt“, kde je napsáno jedno protiopatření. Toto je nevýhodné řešení, protože v případě, že riziko má víc jak jedno protiopatření, tak se musí zkopírovat celý záznam a do toho napsat další protiopatření. Toto se v prototypu řeší snáze skrz „Actions“.

Také to má „termin“, kde toto se dá propojit s „ReviewUntil“, kde tyto položky mají stejný význam.

Zaměřme se teda na hlavní problém – náš prototyp neřeší jeden hlavní problém, nemáme zde dělení skupin rizik. Je možné toto chování nasimulovat s pomocí tagů (štítků), avšak nemá to stejný dopad jako dělení do skupin. Z toho vychází další návrh na novou feature, kde by se přidal další číselník do struktury do databáze pro riziko. Není to řešení, které by vyhovovalo kompletně tomuto testovacímu scénáři, ale částečně by to pomohlo vyřešit.

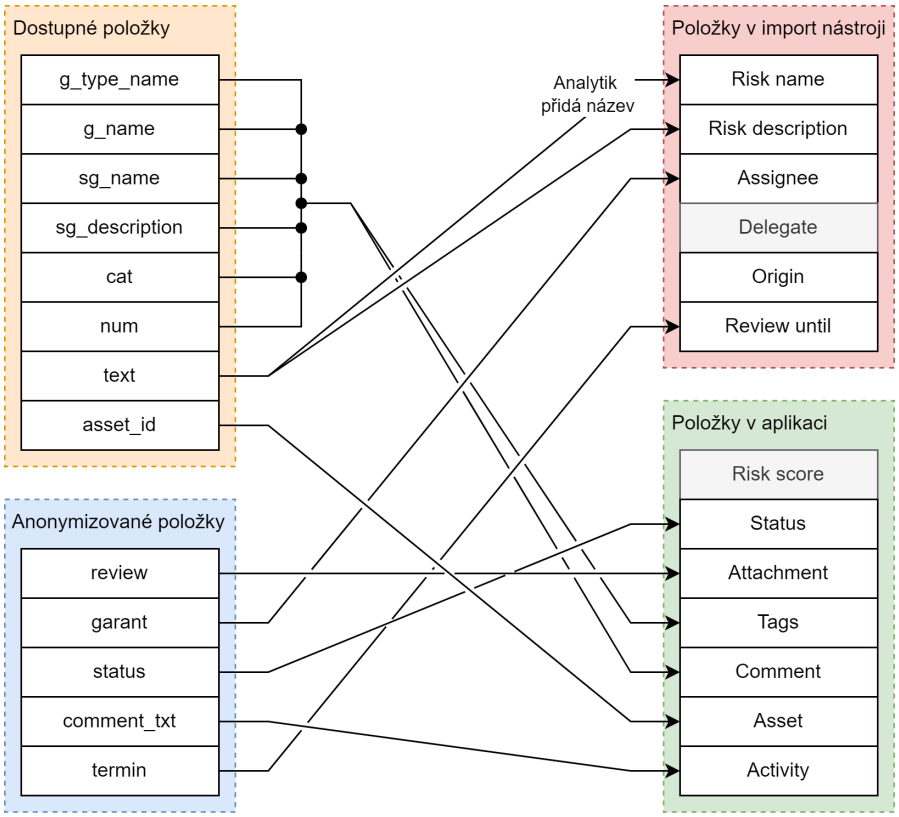
Další část, která byla anonymizovaná, byla review (audit), respektive původ rizika. Tuto část mám reprezentovanou jako origin a také je možné přiložit soubory k riziku, které například ukazují, jak k danému nálezu došlo.

5.1.2 Import dat

Mnou zpracováváný testovací scénář byl v rámci anonymizace oříznutý, zůstaly jenom cizí klíče, respektive ID.

Zde je mnou zvolené mapování položek:

Obrázek 5.2 Ukázka mapování položek



Po dokončení mapování dat připravíme .csv soubor. Na import stránce je jednoduchá ukázka takového souboru. Je zde nutné si dávat pozor na obsah souboru, kde prázdné řádky nevytvoří riziko a

z progress baru (grafický prvek zobrazující načtená rizika) bude vidět tento prázdný řádek.

Toto testování našlo jednu chybu v souboru – není kódovaný s pomocí UTF8. Tím pádem při importu nastane tato chyba:

Obrázek 5.3 Ukázka špatně zakódovaného souboru

2951	Omezen\u0000 pou\u0000\u0000v\u0000n\u0000 syst\u0000mu v ur?it\u0000ch ?asech	Bob	11/5/2024 00:00:00	Unmanaged
2952	Centr\u0000ln\u0000 ov??ov\u0000n\u0000 p?\u0000stupu k syst\u0000mu	Bob		Unmanaged
2953	Povolen\u0000 p?ipojen\u0000 v ur?it\u0000ch ?asech jen v p?\u0000pad? pot?eby	Bob		Unmanaged
2954	Li\u0000\u0000c\u0000 se pracovn\u0000 doby pro jednotliv\u0000 u\u0000ivatele	Bob		Unmanaged
2955	Stanoven\u0000 ?asov\u0000ho pl\u0000nu pro komunika?n\u0000 linky	Bob		Unmanaged
2956	Evidov\u0000n\u0000 aktiv	Bob		Unmanaged
2957	Seznam informa?n\u0000ch aktiv a jejich p?idru\u0000en\u0000ch aktiv	Bob		Unmanaged
2958	Kontrola p?esnosti evidence aktiv	Bob		Unmanaged
2959	Kategorizace informac\u0000 v seznamu informa?n\u0000ch aktiv	Bob		Unmanaged
2960	Vl\u0000cen\u0000 majetku p?ed opu\u0000t?n\u0000m organizace	Bob		Unmanaged

Tuto chybu lze vyřešit jednoduše s pomocí nástroje jako notepad/notepad++, kde se změní encoding na takový, který podporuje českou diakritiku (například UTF-8). Import by pak měl proběhnout v pořádku za podmínky, že ostatní podmínky jsou splněné (riziko nesmí mít prázdný řetězec jako název). Nebo alternativně lze změnit údaje rizika ručně v aplikaci po importu. Není to nejlepší varianta řešení, ale je možné toto řešení provést. Během vlastního testování jsem na tuto chybu nenarazila, protože jsem používala angličtinu.

Další částí importu a testování je provázání dalších položek v aplikaci, tj. Status, přílohy, jestli existují a struktura skupin a kategorií z RAMSES se přenese do tagů. Aktivum se taky donastaví v aplikaci stejně tak jako aktivity.

5.1.3 Práce s riziky po importu

Co se týče další funkcionality aplikace, tak chyběla možnost smazat vícero rizik (například kvůli tomu, že vstupní soubor používal špatný encoding). Z pohledu protiopatření zde byl velký pokrok oproti stávajicímu řešení s pomocí RAMSES.

Po delší době jsem při testování dospěla k další problematické části, která by ideálně měla být při další iteraci dodělána – uživatel by měl mít možnost filtrovat svoje aktivity, kde je „Assignee“ a kde je „Resolver“ (vysvětleno v jiné kapitole).

5.1.4 Závěr z testování

Z testování plyne, že prototyp směřuje správným směrem, avšak chybí mu některé featury, které by se očekávaly u placeného software na řízení rizik. Jinak prototyp splňoval požadavky na řízení rizik a oproti samotnému RAMSES se jedná o znatelný pokrok.

Jako některé z nutných featur, které bych shledala za užitečné, by byl více konfigurovatelný import soubor. Nynější podoba funguje, ale je nutné informace jako asset a status doimplementovávat ručně.

Kapitola 6

Závěr

Bob: „A už jsme připravení na NIS 2 s naší novou aplikací!“
Alice: „Taková aplikace nám vydrží i do NIS 3.“

Na závěr si řekneme, jestli jsou cíle splněné a jaké poznatky můžeme vyvodit z této bakalářské práce.

V rešeršní části a části analýzy řízení rizik bylo provedeno seznámení se s prostředím malých a středních podniků v oblasti kybernetické bezpečnosti, což bylo důležité pro znalost potřeb, které tyto organizace můžou mít v oblasti řízení rizik.

V další části byly analyzovány existující softwarová řešení pro řízení rizik a byla provedena analýza jejich klíčových prodejních argumentů[1]. Na základě této analýzy byl navržen proces řízení rizik, který je navržen pro menší až střední firmy a díky jednoduché škálovatelnosti je možné jej použít i pro velké firmy.

V následující fázi byl navržen a vytvořen prototyp aplikace pro řízení rizik, který umožňuje implementovat navržené procesy a podporuje řízení kybernetických bezpečnostních rizik. Tento prototyp byl dále otestován v reálném prostředí, kde byla ověřena jeho funkčnost a praktická použitelnost. Toto testování pak odhalilo menší nedostatky, které by se pak mohly implementovat v normální verzi.

Nakonec mám tu čest říci, že práce splnila všechny cíle a prototyp, pokud se publikuje pod licenzí MIT, tak bude mít potenciál být vyvíjen dále a může sloužit jako bezplatná alternativa k placeným aplikacím a s podporou komunity být lepší než ostatní software.

Mým jediným problémem s psaním této bakalářské práce byl špatný výběr jazyka, jelikož pojmy již zavedené v angličtině zřídkakdy měly správný pojem v češtině.

[1] Key selling points

Příloha A

Instalační příručka

Pro testování:

- Připravte si MS SQL server a spusťte SQL skript pro vytvoření databáze. Poznamenejte si, jaký máte connection string (tj. způsob, jak se přihlašujete k databázi).
- Skript by neměl doběhnout s chybami, v případě, že nastala chyba, je nutné na chybovou hlášku a vyřešit ji.
- Připravit si Visual studio (2022 a popřípadně vyšší verze), kde otevřete .sln soubor. Zde se musí změnit ConnectionString. V případě, že změníte i název připojení či přidáte nové, pak musíte změnit řetězec ve třídě DB (soubor Data/DB.cs) :

```
public static string cnStr()
{
  return config.GetConnectionString("<Zde dosaďte nový název>");
}
```

- Spuštěním programu lze zjistit, zda program funguje, jak má. V případě, že se nenačítají data, tak zkontrolujte třídu DB nebo ConnectionString, popřípadně step by step debugging.
- Pokud vše funguje jak má, lze tento projekt publikovat a nasadit na webserver.

Bibliografie

1. LUŇÁK, Tomáš. *Úvod do kybernetické bezpečnosti*. 2022. Dostupné také z: https://courses.fit.cvut.cz/BI-UKB/lectures/files/02_Rizika.pdf. BI-UKB, 2. přednáška, ČVUT FIT, Praha.
2. FORTINET. *CIA Triad* [https://www.fortinet.com/resources/cyberglossary/cia-triad]. 2023. Accessed: April 2, 2023.
3. *OneTrust - GRC and Security Assurance Cloud* [https://www.onetrust.com/solutions/grc-and-security-assurance-cloud/]. 2023. Accessed: April 2, 2023.
4. *Qualys - Vulnerability Management, Detection, and Response* [https://www.qualys.com/apps/vulnerability-management-detection-response/]. 2023. Accessed: April 2, 2023.
5. *Access risk management | SailPoint* [https://www.sailpoint.com/products/access-risk-management/]. 2023. Accessed: April 2, 2023.
6. *Hyperproof* [https://sourceforge.net/software/product/Hyperproof/]. 2018. Accessed: April 2, 2023.
7. HARRIS, Shon. *CISSP all-in-one exam guide, 6th edition*. McGraw-Hill, 2013.
8. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 27005: Information technology – Security techniques – Information security risk management*. ISO/IEC, 2011. Dostupné také z: https://www.iso.org/standard/65694.html. ISO/IEC 27005:2011.
9. SOFTEXPERT. *7 methods and tools for risk identification* [https://blog.softexpert.com/en/risk-identification/]. 2021. Accessed:April 2, 2023.
10. BLANK, Rebecca M.; GALLAGHER, Patrick D. *NIST Special Publication 800-30 Revision 1: Guide for Conducting Risk Assessments*. 2012-09. Special Publication, 800-30 (Rev. 1). National Institute of Standards a Technology. Dostupné také z: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf.
11. *NIST National Vulnerability Database (NVD) - CVSS v3 Calculator* [https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator]. 2023. Accessed: April 16, 2023.
12. *MITRE CWE Common Weakness Scoring System (CWSS)* [https://cwe.mitre.org/cwss/cwss_v1.0.1.html]. 2019. Accessed April 16, 2023.
13. CODINGDROPLETS. *Blazor Server Custom Authentication [Blazor Tutorial C# - Part 11]* [https://youtu.be/iq2btD9WufI]. 2022.
14. MICROSOFT. *CircuitOptions.MaxBufferedUnacknowledgedRenderBatches Property* [https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.components.server.circuitoptions.maxbufferedunacknowledgedrenderbatches?view=aspnetcore-7.0]. 2023.

15. RISK ANALYSIS CONSULTANTS. *RAMSES (Risk Analysis and Management System for Enhanced Security)* [https://www.rac.cz/en/ramses-en/]. 2021.

Obsah přiloženého média

```
readme.txt ..... stručný popis obsahu média
src
    impl ..... zdrojové kódy implementace
    thesis ..... zdrojová forma práce ve formátu LaTeX
text ..... text práce
    thesis.pdf ..... text práce ve formátu PDF
```