

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Tvorba webové aplikace pro hlášení škodných událostí
Bakalářská práce

Autor: Jiří Jand'ourek
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Husáková Martina Ph.D.
Odborný konzultant: Ing. Pavel Weigner Ph.D.

Hradec Králové

Duben 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 19.4.2024

vlastnoruční podpis

Jiří Jand'ourek

Poděkování:

Rád bych poděkoval vedoucí bakalářské práce Ing. Martině Husákové Ph.D. za metodické vedení a podporu při zpracování práce.

Dále bych rád poděkoval firmě Cleverbee solutions s.r.o. za možnost podílet se na vývoji produktu Cleeo kterého je tato práce součástí a zejména bych rád poděkoval kolegům Ireně Havlíčkové a Bc. Romanu Soběslavovi za konzultace a cenné rady při návrhu a implementaci aplikace.

Abstrakt

Tato bakalářská práce obsahuje návrh průchodu webovou aplikací. Návrh je vytvořen v programu Figma a je realizován pomocí drátěných modelů (wireframe). Při návrhu byl kladen důraz na jednoduchost a intuitivnost celého procesu vyplňování. Webová aplikace se zaměřuje na sesbírání dat od uživatele pro nahlášení škodní události týkající se především automobilů. Návrh aplikace byl vytvořen na základě existujících aplikací se stejnou tematikou. Samotná aplikace byla i částečně implementována pro demonstraci využití aplikace. Dále byly v práci krátce představeny technologie využití k samotné implementaci aplikace. Využitými technologiemi byly především framework Angular a platforma Liferay. Mimo tyto technologie byly dále využity software Docker pro kontejnerizaci celé aplikace a plugin Yeoman.

Klíčová slova: Škodní událost, Angular, Liferay, Webová aplikace, Frontend, Uživatelské rozhraní

Abstract

Title: Web application development for claims report

This Bachelor thesis is about the designing and developing of a web application. The Web application was firstly designed through Figma web designer tool. The output of this process were several wireframes. The goal of this Bachelor thesis is creating a web application for collecting data for claims reports mainly focusing on cars. Design of this application was based on existing applications with same topic. After designing these, the partial implementation of application followed. There is also brief description of technologies used. The main technologies were the framework Angular and the platform Liferay. Among these technologies were also the user software Docker for containerization of the application and the plugin Yeoman.

Key words: Claims, Angular, Liferay, Web application, Frontend, User Interface

Obsah

1	Úvod.....	1
2	Cíl a metodika práce.....	4
3	Využité technologie	5
3.1	Liferay	5
3.2	Angular.....	6
3.3	Docker	9
3.4	Figma	11
3.5	Yeoman	11
4	Návrh průchodu hlášení škody	12
4.1	Nehoda zahrnující další vozidlo	13
4.2	Vandalismus.....	15
4.3	Krádež	16
4.4	Požár, Exploze, Zkrat.....	17
4.5	Škoda způsobená prostředím	18
4.6	Škoda způsobená zvěří	19
5	Návrh jednotlivých stránek.....	20
5.1	Obecné rozložení.....	20
5.2	Kdy k události došlo.....	22
5.3	Místo události.....	23
5.4	Výběr jedné z možností	24
5.4.1	Kdo nehodu způsobil	24
5.4.2	Co bylo poškozeno	25
5.4.3	Kdo formulář vyplňuje	25
5.5	Výběr z více možností	27
5.5.1	Jak k nehodě došlo	27
5.5.2	Co bylo ukradeno	27

5.6	Informace o vozidle.....	28
5.7	Kde bylo vozidlo poškozeno	30
5.8	Fotografie a dokumenty	31
5.9	Svědci.....	32
5.10	Policejní protokol.....	33
5.11	Platba.....	35
5.11.1	Vozidlo opraveno	35
5.11.2	Vozidlo neopraveno - možnosti	36
5.11.3	Platba po opravě - autorizovaný servis	37
5.11.4	Platba po opravě - bez autorizovaného servisu.....	38
5.11.5	Nejdříve posouzení škody.....	39
5.12	Souhrn.....	41
6	Implementace	43
6.1	Vývoj	43
6.1.1	Stahování Docker Image	43
6.1.2	Vývoj jednotlivých portletů.....	43
6.1.3	Stylování	44
6.1.4	Přidání portletů do pořadí vyplňování	45
6.1.5	Sběr dat	46
6.2	Problémy.....	47
6.2.1	Požadavky na hardware	47
6.2.2	Nasazování - Sdílené balíčky	47
6.2.3	Stylování stránky - Liferay theme	48
7	Shrnutí a diskuse výsledků.....	49
8	Závěry a doporučení	50
9	Seznam použité literatury.....	51

10	Seznam obrázků.....	53
11	Přílohy.....	55

1 Úvod

V dnešní době je snaha o celkovou digitalizaci všech dokumentů a procesů. Tento trend se týká i oblasti pojišťovnictví. Díky digitalizaci lze data rychle a efektivně zpracovávat. To snižuje náklady pojišťovněm a šetří čas zaměstnancům, kteří se tak mohou soustředit na další úkoly. Zároveň i zvyšuje uživatelský komfort a umožňuje vyřešit problém bez nutnosti návštěv pobočky pojišťovny. Tato problematika bude řešena v této práci.

Bakalářská práce byla vytvořena podle zadání od firmy Cleverbee. Části i celá práce byly konzultovány s několika členy týmu pracujícího na projektu Cleeo, jehož je tato práce součástí. Firma Cleverbee se specializuje na digitalizaci agend a dokumentů pro státní správu, zjednodušování a urychlování vnitropodnikových procesů v automobilovém průmyslu, provádí několik projektů pro lokální i světové banky, finanční instituce a také se angažuje v digitalizaci procesů v pojišťovnách – od sjednání pojištění přes B2B systémy až po likvidaci škod.

Produkt Cleeo se zaměřuje na malé pojišťovny, které nemají plně digitalizovány všechny procesy. Tento produkt tak nabídne vše od sjednání nových smluv, správu aktuálních smluv až právě po vypořádání se s následnými škodnými událostmi. Vzhledem k rozsáhlosti se na tvorbě celého produktu Cleeo podílí celý tým a vývoj je rozdělen do několika částí.

Jelikož i v hlášení událostí je obrovský rozsah možností a případných variant, tak se aktuální demo aplikace soustředí pouze na hlášení událostí týkajících se automobilů, a to jak osobních, tak i motocyklů a nákladních aut.

Dále především pro variabilitu koncových zákazníků (pojišťoven) není možno vytvořit specifický vzhled stránek. Proto bude v aktuální fázi produktu pouze navrhnut ideální vzhled pro demonstraci výsledného produktu, ale výsledná implementace bude zahrnovat pouze vzhled prvků pro funkčnost. Dále nebudou

implementovány prvky, které by mohly vyžadovat specifickou konfiguraci od konkrétního zákazníka.

Cílem této části produktu je poskytnout uživatelům jednoduché a přehledné prostředí, které umožní rychle a efektivně vyřešit škodní událost a získat tak potřebné finanční odškodnění. Proto musí uživatel projít několika kroky, které zahrnují výběr pojištěné věci a detailní specifikaci příčiny poškození, až po uvedení místa, času a okolností, za nichž ke škodě došlo. Důležitým prvkem je také specifikování poškozeného majetku a popis toho, jak bude celá situace řešena v případě uznání odškodnění. Aplikace je navržena tak, aby minimalizovat námahu uživatele a současně zajistila, aby veškeré potřebné informace byly poskytnuty srozumitelně a systematicky. Tímto způsobem se uživatelům usnadňuje proces hlášení škodní události a zároveň zajišťuje, aby pojišťovna mohla rychle a efektivně reagovat na jejich žádosti o odškodnění.

Tato práce je vzorem, který bude následně upravován podle požadavků aktuálního trhu, a tudíž bude dále rozvíjena spolu s celým projektem firmou Cleverbee. Tento nový produkt má význam pro firmu, jelikož rozšiřuje portfolio produktů, které firma může nabízet.

V první části práce budou prozkoumány aktuálně existující řešení od pojišťoven, které již nabízejí možnost online hlášení škodních událostí. Cílem je analyzovat tyto systémy a navrhnout, jaká data nejpravděpodobněji budou potřeba pro úspěšné nahlášení události. Dále je potřeba se zaměřit v jakém pořadí a v jakém množství je vhodné tato data sbírat a jakým způsobem je získat od uživatele tak, aby celý proces byl co nejpřívětivější a nejsrozumitelnější. Cílem je navrhnout takový proces, který uživatel zvládne samostatně, aniž by potřeboval pomoc z call centra nebo jiného zdroje.

Proto bude nutné vytvořit průchod celým procesem nahlášení škody, například pomocí drátěných modelů (wireframe), které slouží k ujasnění všech kroků a základního grafického vzhledu aplikace samotné. Tyto modely budou klíčovými

nástrojem pro pochopení uživatelského toku a usnadnění vývoje aplikace. Díky nim bude možné identifikovat případné problémy nebo nedostatky. Například nedostatky v navigaci či uspořádání prvků. Tím bude zajištěna co nejlepší zkušenost z uživatelského pohledu a při používání aplikace pro hlášení škodných událostí.

Uživatelé výsledné aplikace je běžný člověk bez jakýchkoli hlubších IT znalostí, který v dostatečné míře zvládá ovládat mobilní telefon nebo stolní počítač. Aplikace je navržena tak, aby poskytovala uživateli intuitivní prostředí, které je snadno ovladatelné a přístupné i pro ty, kteří nemají zkušenosti s technologiemi. Cílem je, aby každý uživatel mohl snadno a pohodlně hlásit škodné události spojené s vozidlem.

Dále se předpokládá, že uživatel má informace ohledně samotné škodné události a poškozeného vozidla. To znamená, že uživatel ví, co se události týká, a má dostatek informací o poškození vozidla, aby mohl úspěšně vyplnit potřebná data v aplikaci. Předpokládá se, že uživatel má informace například o tom, jaká škoda vznikla, kde a kdy k události došlo. Tyto informace jsou klíčové pro úspěšné a efektivní hlášení škodné události prostřednictvím aplikace.

2 Cíl a metodika práce

Cílem této bakalářské práce je navrhnout a implementovat webovou aplikaci pro hlášení škodných událostí se zaměřením na automobily. Aplikace má vytvořit takové prostředí, aby bylo intuitivní a přehledné pro zákazníky a jednoduše je provedlo celým procesem. V rámci implementace je naprogramovaná taková část, aby demonstrovala cíle a myšlenky aplikace. Při návrhu byl vyhodnocován vliv rozložení těchto prvků na "Use Experience" a jejich možnost zneužití. Návrh se tak snaží minimalizovat tyto problémy.

Metody řešení práce vycházejí z cíle a logické struktury bakalářské práce. Pomocí analýzy aktuálně existujících řešení konkurenčních produktů je vytvářen návrh průchodu hlášení škodných událostí. V rámci teoretické části jsou představeny konkrétní technologie potřebné pro vypracování praktické části, a to zejména technologie Liferay a Angular dále pak Docker a Yeoman. V teoretické části je také krátce popsán nástroj pro tvorbu vizuálního návrhu Figma.

3 Využité technologie

3.1 Liferay

Liferay [1] je open-source platforma pro správu obsahu a tvorbu webových portálů. Příklady těchto portálů jsou intranety, extranety a další webové aplikace. Jedná se o framework, který umožňuje vývojářům vytvářet komplexní webové aplikace. Dále také pomáhá se správou těchto aplikací, s využitím jednoduchého a intuitivního uživatelského rozhraní.

Základní komponentou celého Liferaye jsou portlety. Portlety jsou malé, nezávislé webové aplikace, které mohou být umístěny na webové stránce. Může se jednat o jednoduché portlety, jako je například pouze zobrazení textu, ale i složité portlety jakými jsou interaktivní formuláře či zobrazování grafů.

Liferay také umožňuje vytvářet různé šablony a rozvržení (layouts) pro zjednodušenou správu těchto webových stránek. Ty umožňují jednoduché strukturování a design webových stránek. Což pomáhá snadno vytvářet a upravovat vzhled a rozvržení jednotlivých portálů.

Pro možnost správy jednotlivých webových aplikací má Liferay integrované funkce pro nastavení uživatelských práv a oprávnění. Díky tomu je možné vytvořit několik uživatelských účtů, které mohou spravovat dané webové aplikace. Tímto nastavením práv je zajištěna úprava stránek, aniž by došlo k narušení bezpečnosti. Liferay díky tomu umožňuje organizacím efektivně spravovat přístup k obsahu a funkcím portálu na základě uživatelských rolí a oprávnění.

Liferay umožňuje změnit rozlišení a vzhled portletů podle velikosti zařízení. Díky tomu je snadné upravit styl pro stolní počítače, mobilní zařízení a další.

Díky všem těmto funkcionalitám je Liferay mocnou a flexibilní platformou pro tvorbu a správu obsahu na těchto webových aplikacích a zároveň poskytuje širokou škálu funkcí pro efektivní řízení a využití webových portálů pro různé organizace.

3.2 Angular

Jedná se o open-source framework vytvořený a spravovaný společností Google. Tento framework zásadně změnil přístup k vývoji webových aplikací. Jedná se o framework, který se zaměřuje na tvorbu jednostránkových webových aplikací (Single Page Application – SPA). Také obsahuje nástroje pro efektivní vývoj a správu kódu.

Angular je založen na komponentech pro tvorbu jednoduše škálovatelných webových aplikací. Obsahuje kolekce dobře integrovaných knihoven, které pokrývají širokou škálu funkcí, včetně směrování, správy formulářů, komunikace mezi klientem a serverem a dalších. Je přívětivý k vývojářům, jelikož obsahuje sady vývojářských nástrojů, které pomáhají při vývoji, sestavování, testování a aktualizaci kódu. [2]

Angular je založen na TypeScript jazyku. TypeScript je dle oficiální dokumentace [3] nadstavba JavaScriptu, které k Javascriptu přidává nejen novou syntaxi, ale umožňuje odchyvat chyby ještě v editoru i bez opakovaného kompilování a spouštění psaného kódu.

Dále má větší propojení s moderními IDE tudíž umožňuje automatické doplňování psaných příkazů. Typescript se automaticky konvertuje na JavaScript, a tedy není potřeba řešit u koncového uživatele kompatibilitu. Pokud má uživatel moderní prohlížeč, který zvládne novější verze JavaScriptu zvládne i aplikace napsané v Typescriptu.

Pro práci s Angularem není nutné se zdlouhavě učit nový jazyk. Jedná se o běžnou práci s HTML, CSS a Javascriptem. Pro využití plného potenciálu Angularu je ale nutné pochopit, jak s kódem Angular pracuje. Je vhodné být také seznámen i se základy Typescriptu a například rozumět třídám, které se v JavaScriptu používají.

Jedním z klíčových prvků celého frameworku jsou komponenty [4]. Využitím architektury komponent Angular poskytuje organizovanou strukturu. Díky tomu lze uspořádat projekt do spravovatelných, dobře organizovaných částí, aby byl kód dobře udržovatelný a škálovatelný. Každá komponenta obsahuje takzvaný dekorátor. Jedná se o jakési nastavení dané komponenty. Dále obsahuje TypeScriptovou třídu, která obsahuje samotné chování komponenty (vstup od uživatele, metody, ...). Zde je příklad komponenty z oficiální dokumentace [4]. Část `@Component` je dekorátor, kterou následuje TypeScriptová třída `TodoListItem`

```
// 📄 todo-list-item.component.ts
@Component({
  standalone: true,
  selector: 'todo-list-item',
  template: `<li>(TODO) Read cup of coffee introduction</li> `,
  styles: ['li { color: papayawhip; }'],
})
export class TodoListItem {
  /* Component behavior is defined in here */
}
```

Obrázek 1: Příklad komponenty [4]

Obsah komponent se dá částečně měnit i dynamicky. Stačí navázat daný prvek na požadovaná data a Angular si je automaticky bude aktualizovat. Také je možno například podle stavu dynamických dat ovládat jednotlivé prvky. Zde je příklad, kdy bude tlačítko vypnuté, pokud jsou nějaké čekající údaje. [5]

```
<button [disabled]="hasPendingChanges"></button>
```

Obrázek 2: Příklad navázání tlačítka na data [5]

Také je možnost navázat akci na interakci s prvkem. Může se jednat například o reakci na stisknutí tlačítka. Další příklady využití lze nalézt v oficiální dokumentaci [5].

Angular také umožňuje zobrazovat HTML prvky pouze za určitých podmínek. Tento příklad zobrazuje sekci, která se zobrazí pouze s administrátorskými právy. [6]

```
<section class="admin-controls" *ngIf="hasAdminPrivileges">  
  The content you are looking for is here.  
</section>
```

Obrázek 3: Příklad zobrazení prvku za podmínky [6]

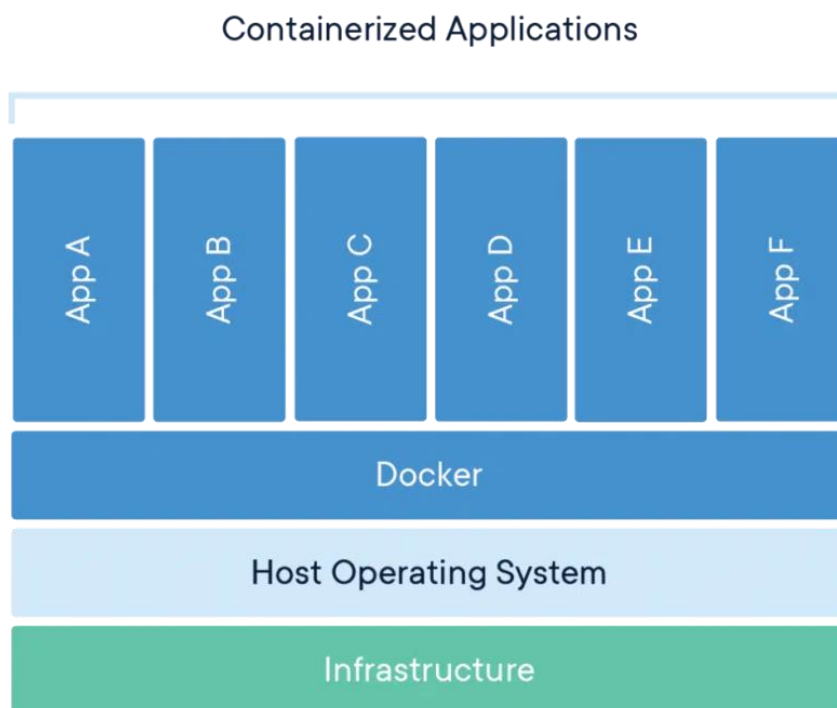
Další užitečnou schopností Angularu je vykreslování seznamů. Zde stačí vytvořit pouze jednu komponentu a Angular ji pomocí seznamu vykreslí tolikrát kolik je dat v seznamu. [7]

```
<ul class="ingredient-list">  
  <li *ngFor="let task of taskList">{{ task }}</li>  
</ul>
```

Obrázek 4: Příklad vykreslování seznamů [7]

3.3 Docker

Docker [8] je aktuálně jeden z nejznámějších softwarů pro kontejnerizaci aplikací. Kontejnerizace je zabalení výsledného programu spolu s knihovnami a závislostmi operačního systému potřebnými ke spuštění aplikace a vytvoření jediného spustitelného objektu (kontejneru) [9]. Díky tomu lze bez obav vyvíjet aplikaci na jedné platformě s jistotou, že bude bez problémů fungovat i na platformě jiné. Kontejner obsahuje pouze požadované aplikace a pro ně specifické soubory, ale neobsahuje (virtualizovaný) operační systém. Tím je výrazně snížena režie na rozdíl od klasických virtuálních strojů. Výhodou Dockeru je proto mnohem menší velikost, větší flexibilita a tím nižší náklady na provoz. Nevýhodou je naopak svázanost s hostujícím operačním systémem, který je přímo využíván pro běh aplikací v kontejnerech. [10]



Obrázek 5: Kontejnerizace [8]

Docker tvoří jednotlivé Docker kontejnery, což jsou ony aplikace se všemi potřebnými knihovnami a závislostmi.

Prvním krokem tvorby Docker kontejneru je vytvoření Dockerfile. Jedná se o textový dokument obsahující příkazy, které mohou být spuštěny v příkazovém řádku při vytváření Image. Také obsahuje cestu k souborům, které jsou potřeba pro správný běh aplikace v kontejneru. Hlavním důvodem tvorby Dockerfile je, že umožňuje automaticky sestavit tzv. Image („obraz“) podle instrukcí, které jsou v něm uloženy.

Docker image je základ kontejneru. Obsahuje seznam všech závislostí a souborů potřebných k dalšímu spuštění kontejneru. Lze ho vytvořit buď z vlastního Dockerfile, nebo nahrát již hotový Image například z Docker Hubu. Docker Hub je oficiální registr obrazů („Docker Images“), kde je možné využívat existující Docker Image, ale i nahrávat vlastní. Image se automaticky nemění. Proto je nutné při změnách na této úrovni vždy znovu sestavit z Dockerfile.

Spuštěný Docker Image se nazývá kontejner. Vlastní software, který má být v kontejneru, a veškeré jeho soubory fungují v tomto kontejneru bez nutnosti dalšího softwaru. Díky tomu, že již vše obsahuje, je velmi snadný pro přenos na další počítače. Software běžící v tomto kontejneru je také oddělený od okolí a tím je minimalizována možnost vzniku chyb.

Komunikace s okolím je omezena. Lze nastavit, které porty budou přístupné. Lze také pracovat se samotným kontejnerem, který můžeme např. vytvářet, spouštět, zastavovat nebo i mazat. Kontejnery lze také vzájemně propojit a vytvořit síť, které jdou opět absolutně oddělit jedna od druhé. Jedná se o takzvané Docker compose.

Pro ukládání dat z kontejneru se používají tzv. Volumes. Jedná se o uložště pro kontejnery, aby si i po vypnutí uložily svá data. Po vypnutí si kontejner ukládá svůj stav. Pokud je potřeba vrátit stav do původní verze, lze kontejner smazat a spustit ho znovu z Image.

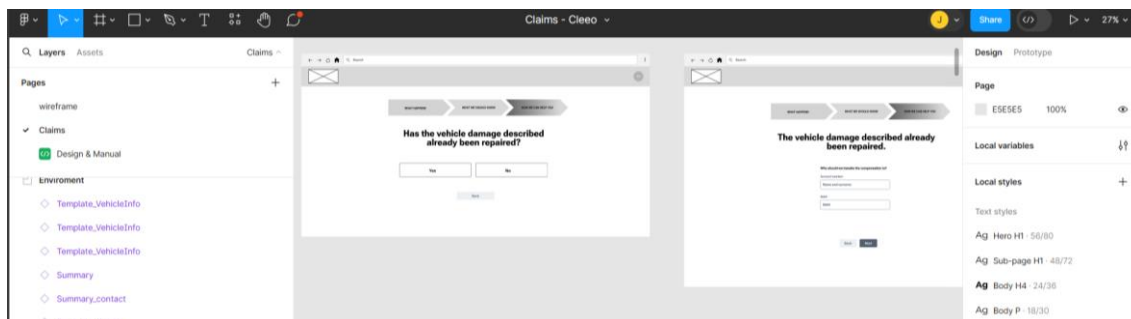
Docker je v projektu využit pouze z důvodu zjednodušení přenosu vyvíjené aplikace mezi jednotlivými pracovními stanicemi.

3.4 Figma

Jedná se o webovou aplikaci sloužící pro navrhování uživatelského rozhraní. Umožňuje spolupráci v reálném čase. Figma [11] nabízí množství funkcí, které umožňují designérům vytvářet, sdílet a spolupracovat na návrzích uživatelských rozhraní. Při návrhu je možné vytvořit i interaktivní prototypování, díky kterému je možné vyzkoušet, jak se bude výsledná aplikace chovat již při jejím navrhování. Je přívětivá i pro vývojáře výsledných aplikací, jelikož umožňuje předpřipravit některé CSS styly.

Díky možnosti tvorby komponentů je jednoduché předpřipravit si některé prvky které lze poté využít opakovaně. Pokud je potřeba vytvořit změnu napříč celým projektem, stačí poté změnit pouze hlavní prvek a veškeré prvky od tohoto zděděné se zaktualizují automaticky.

Nejen při kooperaci v týmu se hodí systém komentářů ale i poznámkových bloků které lze vkládat různě po pracovní „tabuli“.



Obrázek 6: Vzhled aplikace Figma [11]

3.5 Yeoman

Pomocí npm bylo potřeba doinstalovat nástroj Yeoman. Jedná se o plugin, který usnadňuje tvorbu projektů nebo jejich částí. V tomto případě Yeoman sloužil pro tvorbu Angular projektu, který je využit jakožto portlet pro Liferay. [12]

Po doinstalování tohoto nástroje následovalo jednoduché vytvoření Angular projektu pomocí konzolové aplikace. Tato tvorba Angular projektu je velmi intuitivní právě díky tomuto pluginu.

4 Návrh průchodu hlášení škody

Proces navrhování průchodu hlášení škodní události začal analýzou již existujících aplikací od konkurence. S ohledem na vizi firmy, která se zaměřuje na německý trh, byla analýza zaměřena především na dvě konkurenční aplikace, a to pojišťovny ERGO [14] a Allianz [15]. Na základě této analýzy byl vytvořen návrh prostřednictvím wireframů, které pokrývají všechny možné kombinace. Tyto návrhy budou následně implementovány do funkční aplikace.

Bylo nezbytné identifikovat celkem 8 oblastí, ve kterých mohou vzniknout škody na vozidle, a navrhnout intuitivní proces výběru pro uživatele. Tyto oblasti zahrnují:

1. Nehoda zahrnující další vozidlo
2. Nehoda nezahrnující další vozidlo
3. Nehoda způsobená zvěří
4. Poničení skel vozidla
5. Poničení způsobené přírodními vlivy
6. Oheň / exploze / zkrat
7. Škoda způsobená krádeží
8. Škoda způsobená vandalismem

The screenshot shows a user interface for reporting damage. At the top, the title "What happened to you" is displayed in bold, followed by the instruction "Please select the type of damage." Below this, there are eight rectangular buttons arranged in a 2x4 grid. Each button contains a specific type of damage and a small circular icon with a right-pointing arrow. The buttons are: "Accident involving another vehicle", "Accident without the involvement of another vehicle", "Damage from animals", "Glass damage on the vehicle", "Damage from environmental influences", "Fire / explosion / short circuit", "Damage from theft", and "Damage from vandalism". At the bottom center of the screen, there is a "Back" button.

Obrázek 7: Hlavní rozcestník [Vlastní zpracování]

Každá z těchto oblastí představuje specifický typ škody, která může nastat, a je důležité, aby uživatelé měli možnost jednoduše a rychle označit, o jaký typ škody se jedná. Navržený proces výběru prostřednictvím wireframů zajišťuje, že uživatelé budou moci snadno a přesně identifikovat povahu škodní události, což pomůže k rychlému vyřízení pojištění.

4.1 Nehoda zahrnující další vozidlo

Prvním krokem při hlášení nehody s dalším vozidlem je zjištění data a místa události. Vzhledem k tomu, že do události bylo zapojeno další vozidlo, je důležité zjistit, kdo pravděpodobně škodu způsobil. Následně je nutné zjistit, kdo vlastně celou škodu pojišťovně hlásí.

Pokud se jedná o zákazníka pojišťovny, tak není potřeba zjišťovat další detaily, jelikož jsou potřebné informace obvykle dostupné ze smlouvy. Nicméně, pokud hlášení pochází od osoby, která není přímo zákazníkem pojišťovny, a škodu způsobil zákazník pojišťovny, pak je nezbytné získat další údaje o poškozené osobě. Dále uživatel vyplňuje detaily o tom, co vše bylo při události poškozeno a jak k celé události došlo.

Následují informace o vozidlu samotném, jako je značka, model, datum registrace a najetá vzdálenost. Na další stránce se uživateli objeví model vozidla. Na tomto modelu uživatel vyznačí poškozené části a může doplnit i popis poškození, který lze i podpořit fotografiemi a dokumenty na následující stránce. Při vyplňování formuláře se také zaznamenává, zda byl celý incident viděn nějakou třetí osobou. V případě, že ano, je požadován kontakt na tuto osobu, jinak se rovnou přechází na stránku, zda policie o incidentu vede nějaký záznam.

V následující sekci se vyplňuje, jak má v případě vyhovění pojišťovna škodu vyplatit. Pro určení vhodného postupu je nejprve nutné zjistit, zda již bylo poškozené vozidlo opraveno. V případě, že bylo vozidlo již opraveno, uživatel přechází přímo na informace o účtu, na nějž má být částka vyplacena.

Pokud však vozidlo dosud nebylo opraveno, uživatel má na výběr následující možnosti:

- Platba po opravě – autorizovaný servis
- Platba po opravě – bez autorizovaného servisu
- Nejdříve posouzení škody

Při výběru možnosti „Platba po opravě – autorizovaný servis“ je uživatel přeměřován na stránku s výběrem autorizovaných servisů v okolí zadaného PSČ. Po výběru autoservisu se zobrazí informační stránka, jelikož bude uživatel kontaktován přímo vybraným autorizovaným autoservisem, aby se dohodli na podrobnostech opravy a dalším postupu.

V případě, že uživatel zvolí „Platba po opravě – bez autorizovaného servisu“, uživatel zadá pouze informace, kam mají být zaslány peníze.

Když vybere „Nejdříve posouzení škody“, vyžaduje peníze od pojišťovny ještě před započítáním oprav na vozidle. Pojišťovna ale potřebuje posoudit výši škody, proto musí uživatel vybrat, kde k tomuto posouzení dojde. Může si vybrat mezi autorizovaným servisem, kde bude škoda oceněna odborníkem, nebo může určit místo, kam má dojet technik pojišťovny.

Po této sekci se uživatel dostává do poslední části vyplňování celého hlášení, kterou tvoří shrnutí informací, které zadal během celého procesu. Jedná se o tři stránky:

1. První stránka obsahuje souhrn všech informací zadaných uživatelem.
2. Na druhé stránce má uživatel možnost zanechat svojí emailovou adresu pro zasílání novinek o postupu pojišťovny v celém případě.
3. Na poslední stránce se nachází pouze informativní text, který popisuje, jak bude pojišťovna nadále postupovat. Z této stránky se uživatel může vrátit na domovskou stránku pojišťovny. Tím se uzavírá proces hlášení škody.

4.2 Vandalismus

Prvním krokem v případě vandalismu je zjistit, kdy k poškození došlo a kde vozidlo v této době bylo. Vzhledem k tomu, že osobní automobil není jediným vozidlem, které pojišťovny pojišťují, je nezbytné také zjistit, které vozidlo bylo poškozeno a které části vozidla byly poškozeny. Toho docílíme výběrem z typů vozidel jako jsou nákladní automobily, osobní automobily, motorky a další.

Poté uživatel popisuje, které části byly poškozeny. V případě, že se nejedná o osobní automobil, se model vozidla nezobrazuje a uživatel musí popis škody vyplnit do textového pole. Následně je potřeba zjistit model vozidla. Vzhledem k pouze částečnému propojení německých registrů vozidel, není vždy možné pomocí SPZ jednoduše dohledat veškeré informace o vozidle. Proto následují stránky, kde uživatel vyplňuje značku, model vozidla a datum registrace, nebo má možnost zadat VIN kód, případně HSN/TSN označení vozidla.

Pro zjednodušení odhadu aktuální ceny vozidla uživatel zadává na této stránce i přibližný stav tachometru. Toto opatření je důležité pro přesnější odhad hodnoty vozidla. V případě, že byli vandalismu přítomni svědci, je uživatel na další stránce vyzván, aby zadal kontaktní údaje na tyto osoby. Většina pojišťoven vyžaduje, aby o události vedla záznam policie. Pokud má uživatel k dispozici číslo policejního spisu, zadává ho na příslušné stránce. Pokud takový záznam policie neexistuje, uživatel je vyzván, aby vandalismus nahlásil policii, avšak může pokračovat i bez tohoto úkonu a vyplnit formulář dále.

Po zadání všech nezbytných informací může uživatel nahrát fotografie a dokumenty týkající se události, což je důležité pro dokumentaci události. Poté může přejít na další sekvenci stránek, kde je pojišťovně nahlášeno, kam mají být peníze zaslány, případně jak bude vozidlo opraveno. Nakonec se uživateli zobrazí souhrn celého formuláře a informace o tom, jak bude dále pojišťovna postupovat. Tento přehled zajišťuje uživateli jasné pochopení toho, co následuje po vyplnění formuláře, a pomáhá mu udržet přehled o celém procesu správy škodní události.

4.3 Krádež

Jak již název naznačuje, tímto procesem projde uživatel, pokud mu bylo jeho vozidlo, ať už celé nebo jen jeho část, odcizeno. V případě, že došlo k odcizení celého vozidla, uživatel je automaticky přesměrován na stránku, kde specifikuje, kdy k odcizení došlo. V opačném případě, kdy byla odcizena pouze část vozidla, je uživatel nejprve vyzván k detailní specifikaci, které konkrétní části byly odcizeny, a až poté přechází na stránky, kde udává informace o čase a místě odcizení. Následně oba scénáře postupují stejně.

Uživatel je požádán k poskytnutí dalších informací o vozidle, jako je výrobce, model, nebo VIN, případně HSN/TSN a stav tachometru. Dále se zjišťuje, zda je o případu veden záznam u policie, a pokud ano, zadává se číslo policejního spisu. Kromě toho má uživatel možnost nahrát dodatečné dokumenty a fotografie, které mohou sloužit pro vyhodnocování škodní události.

Po těchto krocích se opět varianty rozcházejí v závislosti na tom, zda byla ukradena část nebo celé vozidlo. V případě, že došlo k odcizení celého vozidla, uživatel přechází rovnou na souhrn informací z celého formuláře. Naopak, v případě krádeže pouze části vozidla, je uživatel vyzván k specifikaci, kam má pojišťovna zaslat peníze, nebo jak bude vozidlo opraveno, a je informován o tom, jak bude škoda na vozidle vyhodnocena. Teprve poté následuje přechod na souhrn informací z celého formuláře. Tímto způsobem je proces přizpůsoben konkrétní povaze škodní události.

4.4 Požár, Exploze, Zkrat

Prvním krokem je vybrat která z možností nastala. V tomto případě to neovlivňuje žádný z dalších kroků, avšak je důležité zaznamenat povahu události pro pojišťovnu. Následuje sled otázek týkajících se času a místa incidentu. Uživatel je vyzván k poskytnutí informací o tom, kde a kdy se událost stala.

Dále je nutné vyplnit, jaký typ vozidla byl poškozen, a vyznačit, které části vozidla byly poškozeny, případně poskytnout popis poškození. Další informace o vozidle jako je značka, model, VIN nebo HSN/TSN, zadává uživatel na stránce následující hned po popisu škody na vozidle a před stránkou, zda policie vede záznam o této události. Pokud policie záznam nevede, je uživatel vyzván, aby celou událost policii nahlásil, nebo aby případně dodal číslo policejního spisu.

Poté následuje čas pro nahrání fotografií a dokumentů od uživatele a přechází se na sběr informací o tom, kam má pojišťovna zaslat peníze, nebo jak bude vozidlo opraveno. Nakonec se uživateli zobrazí shrnutí všech informací, které zadal při vyplňování formuláře, což mu umožňuje přehledně zkontrolovat zadané údaje a zajišťuje, že jsou všechny potřebné informace správně dokumentovány pro další správu škodní události.

4.5 Škoda způsobená prostředím

Jedná se o velmi podobný postup jako u Požár, Exploze, Zkrat. Proto v prvním kroku uživatel vyplňuje, jak byla škoda vlastně způsobena. Uživatel je vyzván k výběru z velkého množství možností, jelikož škoda může být způsobena různými přírodními živly. Mezi tyto živly patří například bouře, blesky, záplavy, sesuvy půdy, a dokonce i vulkanické erupce.

Poté se pokračuje poměrně stejnou sekvencí stránek, která začíná otázkami týkajícími se času a místa události. Dále o jaký typ vozidla se jednalo, následovaný vyznačením na modelu, kde bylo vozidlo poškozeno, případně popis poškozené oblasti. Poté informace o vozidle jako je značka, model případně VIN nebo HSN/TSN. Je zde také vyhrazen prostor pro dokumenty a fotografie od uživatele.

Dále pro pojišťovnu uživatel musí vyplnit sled otázek, na základě kterých se rozhoduje, kam má být pojistka vyplacena. Jedná se o otázky, zda bylo vozidlo již opraveno, případně kam mají být peníze odeslány. Pokud vozidlo ještě opraveno nebylo, je uživateli nabídnuto několik možností. Buď nechat vůz opravit v autorizovaném servisu a případně vybrat konkrétní autorizovaný servis, nebo nechat vozidlo opravit v jiném než autorizovaném servisu, a tudíž určit, kterému autoservisu mají být peníze zaslány. Poslední možností je nechat si vyplatit peníze ještě před opravením vozidla. Při této volbě je nutné, aby byla odborně odhadnuta škoda způsobená na vozidle. To lze zařídit dvěma způsoby: kontrolou v autorizovaném servisu, nebo na konkrétní adrese určené uživatelem.

Po vyplnění těchto informací je uživateli zobrazen souhrn všech údajů zadaných během vyplňování formuláře. Zde má uživatel možnost zanechat svoji emailovou adresu pro odesílání novinek ohledně postupu vyřizování žádosti. Následuje informační stránka, kde je uživatel informován o dalším postupu pojišťovny. Tento souhrn informací umožňuje uživateli přehledně zkontrolovat všechny zadané údaje a zajišťuje, že jsou všechny potřebné informace správně zaznamenány pro další postup při řešení škodní události.

4.6 Škoda způsobená zvěří

V tomto případě se nemusí jednat jen o srážku se zvěří, ale také o poškození způsobené například rozkousáním kabelů. Na to, jak k události došlo, se ptáme hned v prvním kroku, a to volbou mezi „Srážka se zvěří“ nebo „Kousnutí kunou či jiným zvířetem“.

Kdy k události došlo se zjišťuje hned v dalším kroku, následuje dotaz -, kde k události došlo. Dále se zjišťuje, jak k události došlo a o jaké vozidlo se jedná následováním podrobností o vozidlu. Poté se zde nachází model automobilu pro výběr poškozené části vozidla.

Pokud se uživatel rozhodne nahrát fotodokumentaci, nebo jiné dokumenty, může tak učinit na další stránce před stránkou se zadáváním policejního spisu.

Dále musí uživatel odpovědět na sérii otázek, které pomáhají pojišťovně určit, kam má být pojistka vyplacena. Pokud vozidlo ještě nebylo opraveno, uživateli se nabízejí různé možnosti. Může si vybrat, zda chce nechat vůz opravit v autorizovaném servisu a pokud ano, může vybrat konkrétní autorizovaný servis, nebo pokud preferuje jiný autoservis, může určit, kam mají být peníze zaslány. Poslední možností je požádat o vyplacení peněz ještě před opravením vozidla, ale v tomto případě musí být odborně odhadnuta výše škody. To lze provést buď prostřednictvím odborné kontroly v autorizovaném servisu, nebo na konkrétní adrese, kterou určí uživatel.

Po vyplnění těchto informací je uživateli zobrazen souhrn všech údajů, které zadal během vyplňování formuláře. Také má možnost zanechat svou emailovou adresu pro zasílání aktualizací ohledně postupu vyřizování jeho žádosti. Následuje informační stránka, která uživatele informuje o dalším postupu pojišťovny.

5 Návrh jednotlivých stránek

V této sekci budou shrnuty návrhy stránek, na které může uživatel při hlášení škodní události narazit. Pro jednodušší pročitání této práce se v následujících sekcích mohou nacházet oříznuté a zmenšené. Stránky v originálním vizuálu se nacházejí v příloze.

5.1 Obecné rozložení

Vzhledem k tomu, že se jedná o webovou aplikaci, musíme počítat s prohlížečovým oknem. Toto okno většinou zahrnuje pole pro vyhledávání, tlačítka pro navigaci (domů, vpřed a zpět, znovunačtení stránky) a další prvky, které nelze ovlivnit.

Pod lištou prohlížeče se nachází prostor naší aplikace, kde bude umístěno menu. To bude obsahovat logo aplikace, které umožní uživatelům vrátit se zpět na domovskou stránku, a také odkaz na uživatelský profil (pokud je uživatel přihlášen, některé otázky na uživatele v průběhu vyplňování budou automaticky vyplněny). Toto menu bude statické a při procesu hlášení události se nebude měnit.

Pro zobrazení pokroku ve vyplňování formuláře je v každém portletu aplikace přidán pruh, který obsahuje stručné shrnutí celého postupu. Tento pruh je rozdělen do tří částí:

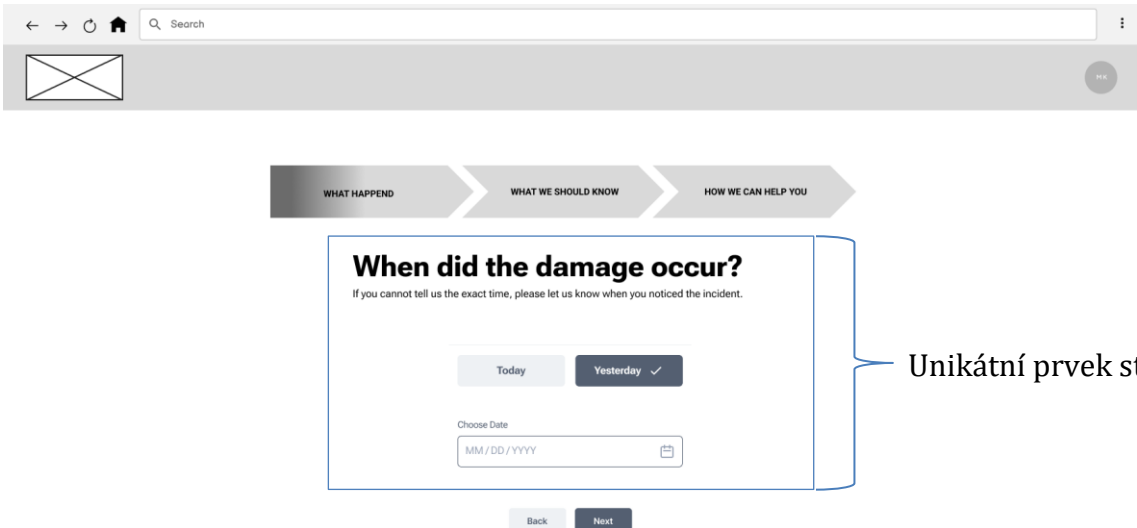
1. Co se stalo: Tato část obsahuje informace o události, včetně popisu toho, co se přesně stalo a detailů o zapojených vozidlech.
2. Co bychom měli vědět: Zde uživatel doplňuje další informace k události, které mohou být důležité pro pojišťovnu.
3. Jak vám můžeme pomoci: Tato část se zabývá rozhodnutím, zda bude škoda na vozidle opravena nebo proplacena.

Tento pruh nebude v implementaci realizován, jelikož přesný postup a stav vyplňování bude upraven pro každého koncového zákazníka (pojišťovnu).

Dále se na každé stránce nachází nadpis aktuálního kroku a případně jeho popis, aby bylo uživatelům jasné, kde se právě nacházejí a co je potřeba vyplnit. Tento prvek se sice nachází na každé stránce ale jeho obsah závisí na aktuální stránce.

Posledním prvkem který se na většině stránek opakuje je tlačítko pro přesun mezi jednotlivými kroky. Většinou se jedná o dvojici tlačítek „další“ a „zpět“. Nicméně, v závislosti na aktuální situaci na stránce, může být tlačítko "další" nahrazeno jinou variantou, jako například tlačítkem pro odeslání hlášení o škodě na konci vyplňování formuláře.

Následující obrázek je plným zobrazením stránky a tudíž i všech prvků těchto prvků.

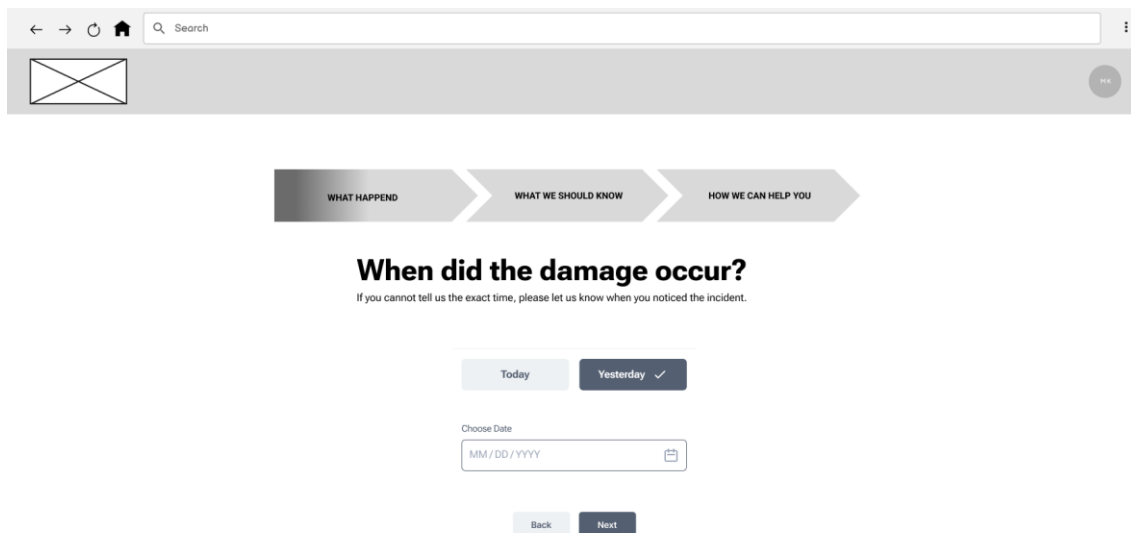


The image shows a browser window displaying a form with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The current step is 'When did the damage occur?' with the instruction: 'If you cannot tell us the exact time, please let us know when you noticed the incident.' The form includes radio buttons for 'Today' and 'Yesterday' (selected), a 'Choose Date' field with a calendar icon, and 'Back' and 'Next' buttons. A blue bracket on the right highlights the form content, labeled 'Unikátní prvek stránky'.

Obrázek 8: Kdy k události došlo? [Vlastní zpracování]

5.2 Kdy k události došlo

Kromě opakujících se prvků se na této stránce nachází pole určená k vyplnění data, kdy k události došlo. Po rozkliknutí tohoto pole má uživatel možnost zadat datum ručně, ale také se mu zobrazí kalendář (datepicker), což usnadňuje pohodlné vyplnění.



The screenshot shows a web browser window with a search bar and a navigation bar. Below the navigation bar is a progress indicator with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The current step is 'When did the damage occur?'. Below the title is a sub-header: 'When did the damage occur?' and a note: 'If you cannot tell us the exact time, please let us know when you noticed the incident.' The form has two buttons: 'Today' and 'Yesterday' (selected). Below these is a 'Choose Date' section with a text input field containing 'MM/DD/YYYY' and a calendar icon. At the bottom are 'Back' and 'Next' buttons.

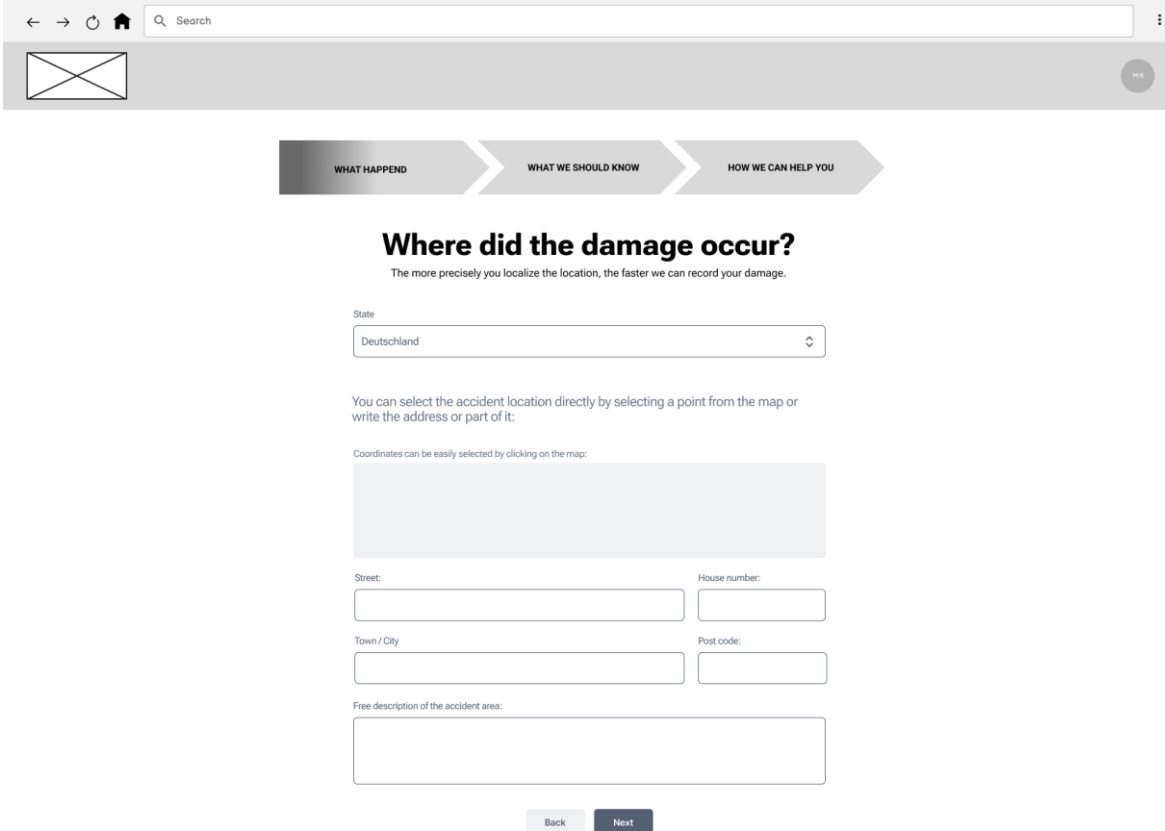
Obrázek 9: Kdy k události došlo? [Vlastní zpracování]

Také jsou zde umístěna tlačítka umožňující rychlý výběr dnešního a včerejšího data. Toto uspořádání je navrženo s ohledem na předpoklad, že uživatel bude událost hlásit pojišťovně v co nejbližší možné době po samotném incidentu.

5.3 Místo události

Vzhledem k tomu, že se incident může stát i v zahraničí, je nutné nejdříve specifikovat zemi. Při výběru bude uživatel přesunut na následující mapce na uvedený stát, kde si může vybrat danou adresu. Pokud si uživatel vybere na mapce, jsou následující údaje automaticky doplněny. V případě, že uživatel nevybere místo události na mapě, je nutné adresu zadat manuálně vyplněním všech polí.

Poslední pole slouží k doplnění jakýchkoli podrobností, které uživatel uzná za vhodné. Tímto zajišťujeme, že uživatel má možnost poskytnout veškeré relevantní informace o místě události.



The screenshot shows a web interface for reporting damage. At the top, there is a browser address bar and a search field. Below it is a progress bar with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The current step is 'Where did the damage occur?'. Below the title, there is a sub-header: 'The more precisely you localize the location, the faster we can record your damage.' The form includes a 'State' dropdown menu with 'Deutschland' selected. Below this, there is a text box for selecting the accident location directly by selecting a point from the map or writing the address or part of it. A note states: 'Coordinates can be easily selected by clicking on the map:'. Below this is a large, empty rectangular area representing a map. The form also includes four input fields: 'Street', 'House number', 'Town / City', and 'Post code'. At the bottom, there is a 'Free description of the accident area:' label and a large text area. Finally, there are 'Back' and 'Next' buttons.

Obrázek 10: Kde k události došlo? [Vlastní zpracování]

5.4 Výběr jedné z možností

V průběhu vyplňování formuláře je často potřeba vybrat z několika možností. Proto je navrhnut jednotný styl pro tento výběr, který usnadňuje uživatelům orientaci a plynulost vyplňování.

Po vybrání dané možnosti je uživatel automaticky přesunut na další stránku, aniž by musel kliknout na tlačítko "další", případně se mu zobrazí pokračování dané stránky jako v případě [5.4.3 Kdo formulář vyplňuje](#). Ve formuláři se tento styl vyskytuje opakovaně, přičemž v této podkapitole jsou zobrazeny pouze některé z možností.

5.4.1 Kdo nehodu způsobil

Zde se jedná spíše o zjednodušený odhad, jelikož finální informací, kdo vlastně nehodu způsobil, je policejní záznam o nehodě. Tento záznam je většinou požadován později při vyplňování.

The image shows a mobile application interface. At the top, there is a navigation bar with a search bar and a home icon. Below the navigation bar, there are three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The main heading is 'Who do you think caused the damage?'. Below the heading, there is a prompt: 'Please indicate who you think caused the damage.' There are three options, each with a car icon and the text 'VEHICLE INSURANCE':

- I
- Accident opponent
- Accident opponent and I

At the bottom, there is a 'Back' button.

Obrázek 11: Kdo nehodu způsobil? [Vlastní zpracování]

5.4.2 Co bylo poškozeno

V této sekci uživatel vybírá, který druh vozidla byl poškozen. Uživatel má možnost specifikovat, zda se jednalo o osobní automobil, motocykl, nákladní vůz nebo jiný typ vozidla.

The screenshot shows a web browser interface with a search bar and navigation icons. Below the browser is a progress bar with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The main content area is titled 'What has been damaged?' and includes the instruction 'Please indicate here who or what was damaged (multiple selection possible)'. There are four input fields, each with a car icon and the text 'VEHICLE INSURANCE' below it. The options are 'Car', 'Truck', 'Motorcycle', and 'Others'. A 'Back' button is located at the bottom of the form.

Obrázek 12: Co bylo poškozeno? [Vlastní zpracování]

5.4.3 Kdo formulář vyplňuje

V této části uživatel udává, zda je vůbec klientem dané pojišťovny. Může se stát, že osoba vyplňující tento formulář není přímo tím, kdo má sjednané pojištění, ale spíše osobou, která žádá o vyplacení škody způsobené klientem pojišťovny.

What applies to you?

Are you insured with us or would you like to report damage as an opponent of an accident?

The screenshot shows a web form titled 'What applies to you?' with the question 'Are you insured with us or would you like to report damage as an opponent of an accident?'. There are two input fields, each with a car icon and the text 'VEHICLE INSURANCE' below it. The options are 'I am your customer' and 'Your customer caused me the damage'.

Obrázek 13: Kdo formulář vyplňuje? [Vlastní zpracování]

Pokud uživatel označí možnost "Jsem váš zákazník", bude automaticky přesměrován na další stránku. Avšak pokud vybere druhou možnost "Váš zákazník mi způsobil škodu", zobrazí se mu další část formuláře, kde může poskytnout další informace o sobě.

← → ↻ 🏠 🔍 Search

WHAT HAPPEND WHAT WE SHOULD KNOW HOW WE CAN HELP YOU

What applies to you?

Are you insured with us or would you like to report damage as an opponent of an accident?

I am your customer

VEHICLE INSURANCE

Your customer caused me the damage

Please enter your license plate number.

Salutation:

Men Firm

Enter injured person

First Name Surname

Street and house number House number:

Town / City Post code:

State

Country

Email address Mobile

Please enter the license plate number of our policyholder.

Back Next

Obrázek 14: Kdo formulář vyplňuje? – rozšířeno [Vlastní zpracování]

5.5 Výběr z více možností

Na rozdíl od předcházející kapitoly si zde uživatel může z daných možností vybrat více než jednu možnost.

5.5.1 Jak k nehodě došlo

V této části formuláře má uživatel možnost vybrat jednu nebo více příčin, které vedly k celé události. Nabízené možnosti obsahují různé příčiny, které mohou zahrnovat například zatáčení, parkování, změna jízdního pruhu nebo nedání přednosti v jízdě. Uživatel má možnost vybrat všechny příčiny, které se týkají dané události, a poskytnout tak pojišťovně co nejpresnější informace o tom, co se stalo.

How did the damage come about?

Please give us details of how the accident happened, vehicle damage will be recorded later.
At the end of the form you can upload photos and other documents.

- Tailgating
- Maneuvering /parking
- Disregard for the right of way
- Turn
- Off the roadway
- Overtaking
- Lane change
- Miscellaneous

Obrázek 15: Jak k nehodě došlo? [Vlastní zpracování]

5.5.2 Co bylo ukradeno

Pokud došlo k situaci, kdy bylo vozidlo či jeho část ukradeno, uživatel je vyzván k vyplnění podrobností o ukradených věcech.

What was stolen?

- Antenna
- Side Mirror
- Wheels
- Airbag(s)
- Audio device(s)
- Sat Nav(s)
- Other

Obrázek 16: Co bylo ukradeno? [Vlastní zpracování]

5.6 Informace o vozidle

Je nutné zjistit co nejvíce informací o poškozeném vozidle pro přesné určení výše případné škody. V každém případě je nutné dodat státní poznávací značku (SPZ) daného vozidla. Na německém trhu, kam se firma zaměřila, není dobře propojen registr vozidel, což brání snadné identifikaci konkrétního vozidla prostřednictvím SPZ pro účely pojišťovny. Proto je požadováno, aby uživatel ručně vyplnil informace o výrobci, modelu a datu registrace vozidla. Alternativně má možnost uvést VIN (Vehicle Identification Number) kód, který poskytuje unikátní identifikaci vozidla. Poslední možností je použití HSN a TSN (Hersteller und Typschlüsselnummer) kódu pro identifikaci konkrétního vozidla. Jakoukoli z těchto informací pojišťovna využije k identifikaci poškozeného vozidla. Dále je uživatel požádán o vyplnění najeté vzdálenosti vozidla a o informaci, zda je vozidlo v soukromém vlastnictví. Tyto údaje jsou důležité pro správné vyčíslení škody a stanovení dalšího postupu pojišťovny.

The screenshot shows a web browser interface with a search bar and navigation icons. Below the browser is a navigation bar with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The main content area is titled 'Can you provide more information about the damaged vehicle?' and contains the following form elements:

- A search bar with a dropdown menu.
- Buttons for 'Search', 'VIN', and 'HSN/TSN'.
- A question: 'What brand and model is it?' with two dropdown menus for 'Brand' and 'Model'.
- A question: 'When was the vehicle first registered?' with a date input field (MM/DD/YYYY).
- A question: 'What is the mileage of the damaged vehicle?' with three radio button options: 'Less than 100,000 km' (selected), '100,000 - 200,000 km', and 'More than 200,000 km'.
- A question: 'Is this a private vehicle?' with two radio button options: 'Yes' (selected) and 'No'.
- 'Back' and 'Next' buttons at the bottom.

Obrázek 17: Informace o vozidle – Značka a model [Vlastní zpracování]

Can you provide more information about the damaged vehicle?

VIN code:

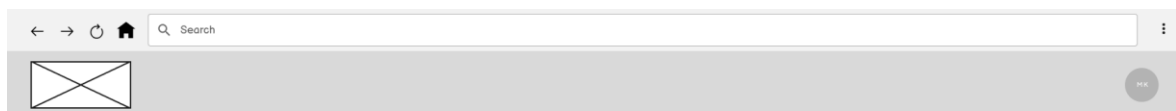
What is the mileage of the damaged vehicle?

- Less than 100,000 km
 100,000 - 200,000 km
 More than 200,000 km

Is this a private vehicle?

Yes No

Obrázek 18: Informace o vozidle – VIN [Vlastní zpracování]



WHAT HAPPEND

WHAT WE SHOULD KNOW

HOW WE CAN HELP YOU

Can you provide more information about the damaged vehicle?

HSN code:

TSN code:

What is the mileage of the damaged vehicle?

- Less than 100,000 km
 100,000 - 200,000 km
 More than 200,000 km

Is this a private vehicle?

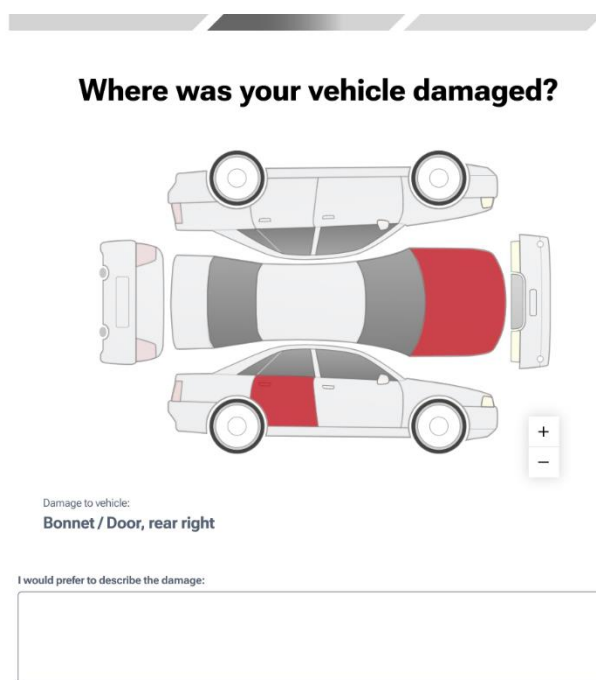
Yes No

Obrázek 19: Informace o vozidle – HSN/TSN [Vlastní zpracování]

5.7 Kde bylo vozidlo poškozeno

Na této stránce uživatel zaznamenává, které části vozidla byli poškozeny. V případě osobního automobilu se zobrazí model vozu, ze kterého si uživatel může vybrat poškozené části.

Pokud se bude jednat o jiné vozidlo než osobní automobil objeví se pouze místo pro slovní popis do kterého uživatel škodu slovně detailněji popíše.

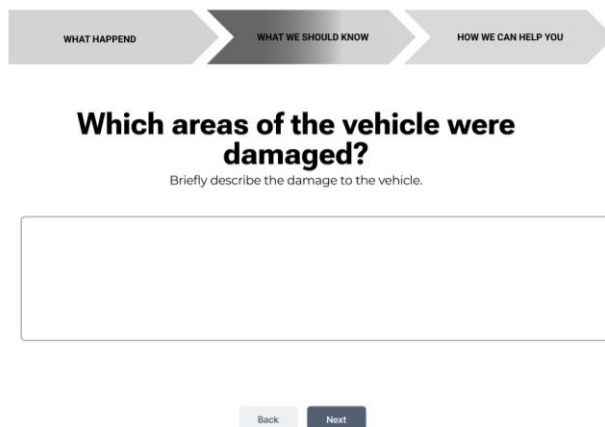


Where was your vehicle damaged?

Damage to vehicle:
Bonnet / Door, rear right

I would prefer to describe the damage:

Obrázek 20: Kde bylo vozidlo poškozeno? – model [Vlastní zpracování]



WHAT HAPPEND WHAT WE SHOULD KNOW HOW WE CAN HELP YOU

Which areas of the vehicle were damaged?

Briefly describe the damage to the vehicle.

Back Next

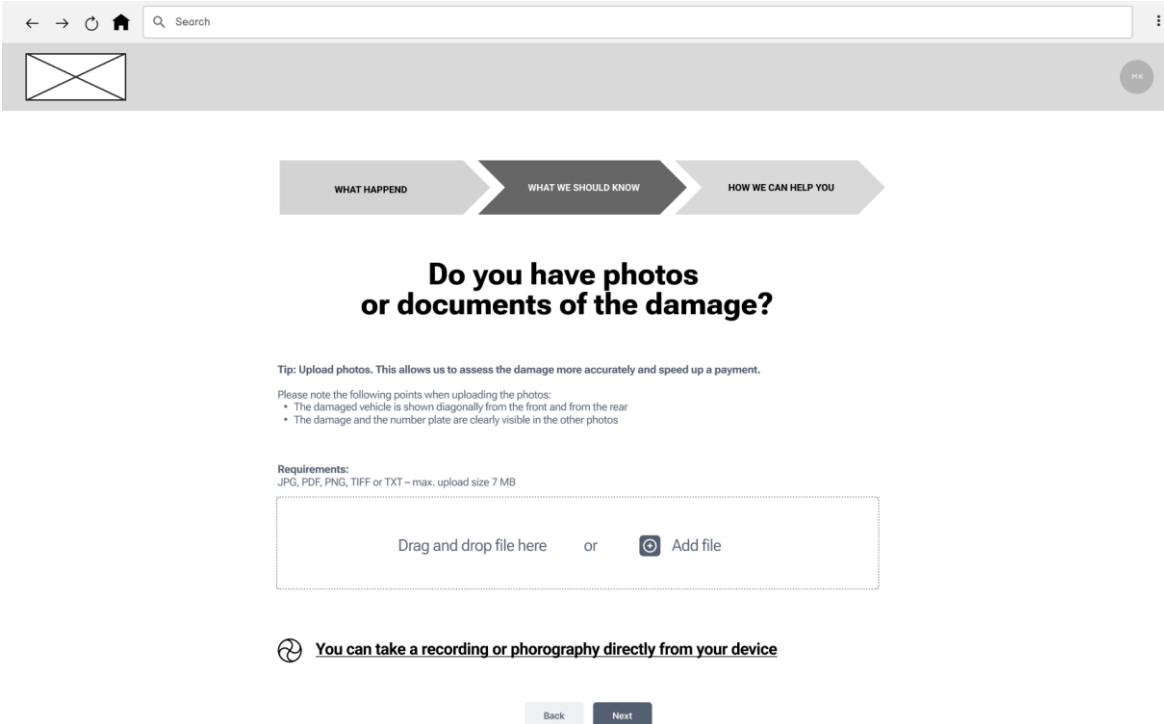
Obrázek 21: Kde bylo vozidlo poškozeno? – textové pole [Vlastní zpracování]

5.8 Fotografie a dokumenty

Na této stránce má uživatel možnost nahrát veškeré dokumenty a fotografie, které by mohly být pro pojišťovnu užitečné při vyřizování škodní události.

Uživatel může jednoduše nahrát soubory přetažením do určené oblasti na stránce nebo kliknutím na tlačítko, které otevře průzkumník souborů na jeho zařízení.

Tímto způsobem je umožněno uživateli snadno a pohodlně přidat veškeré relevantní dokumenty a fotografie k vyplňovanému formuláři.



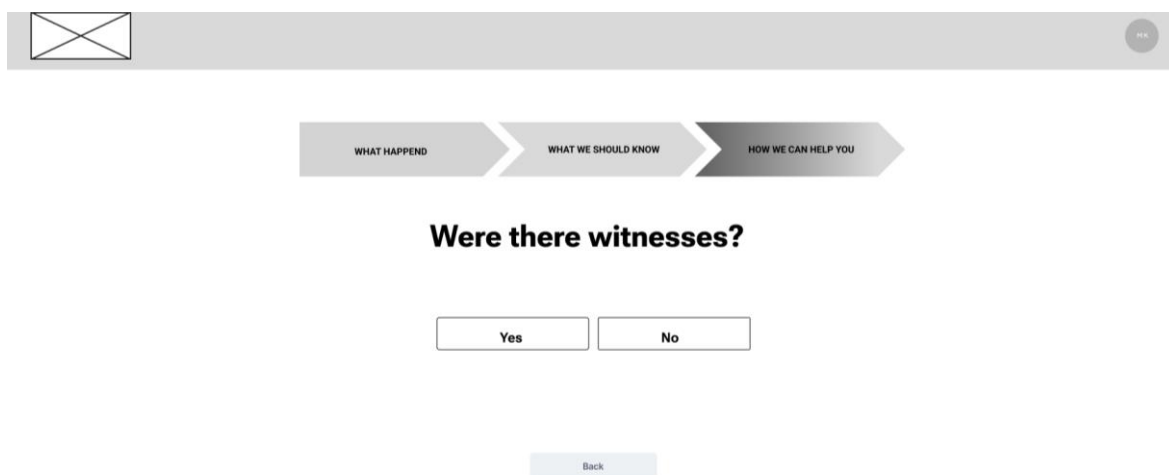
The screenshot shows a web browser window with a search bar and a home icon. Below the browser is a navigation bar with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The main content area is titled 'Do you have photos or documents of the damage?'. It includes a tip: 'Tip: Upload photos. This allows us to assess the damage more accurately and speed up a payment.' Below the tip are instructions: 'Please note the following points when uploading the photos:' followed by two bullet points: '• The damaged vehicle is shown diagonally from the front and from the rear' and '• The damage and the number plate are clearly visible in the other photos'. Below the instructions are requirements: 'Requirements: JPG, PDF, PNG, TIFF or TXT – max. upload size 7 MB'. The main upload area is a dashed box containing the text 'Drag and drop file here' and 'or' followed by a camera icon and 'Add file'. Below the upload area is a camera icon and the text 'You can take a recording or photography directly from your device'. At the bottom are 'Back' and 'Next' buttons.

Obrázek 22: Fotografie a dokumenty [Vlastní zpracování]

5.9 Svědci

Na této stránce je uživatel nejdříve dotázán, zda byli incidentu přítomni nějakí svědci. Pokud nebyl žádný svědek přítomen, uživatel může tento krok přeskočit a pokračovat dál v procesu.

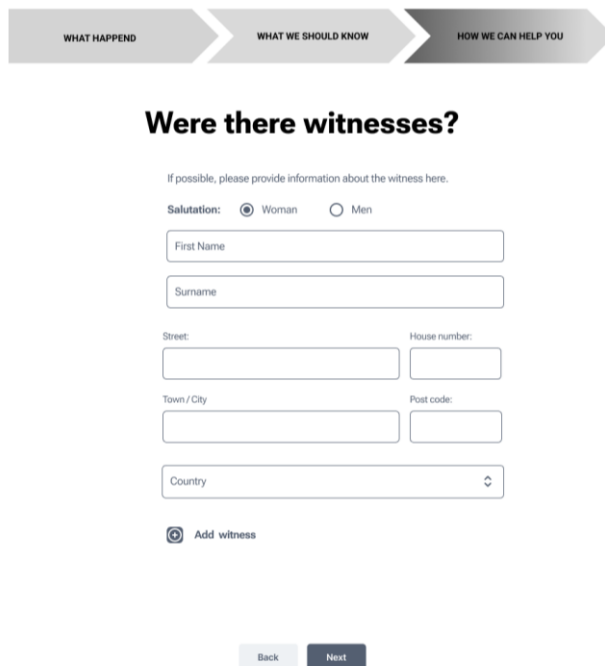
V případě, že svědkové byli přítomni, je uživatel požádán, aby vyplnil veškeré dostupné informace o těchto svědcích. To zahrnuje jejich jméno a kontaktní údaje.



WHAT HAPPEND WHAT WE SHOULD KNOW HOW WE CAN HELP YOU

Were there witnesses?

Obrázek 23: Svědci [Vlastní zpracování]



WHAT HAPPEND WHAT WE SHOULD KNOW HOW WE CAN HELP YOU

Were there witnesses?

If possible, please provide information about the witness here.

Salutation: Woman Men

First Name

Surname

Street: House number:

Town / City Post code:

Country

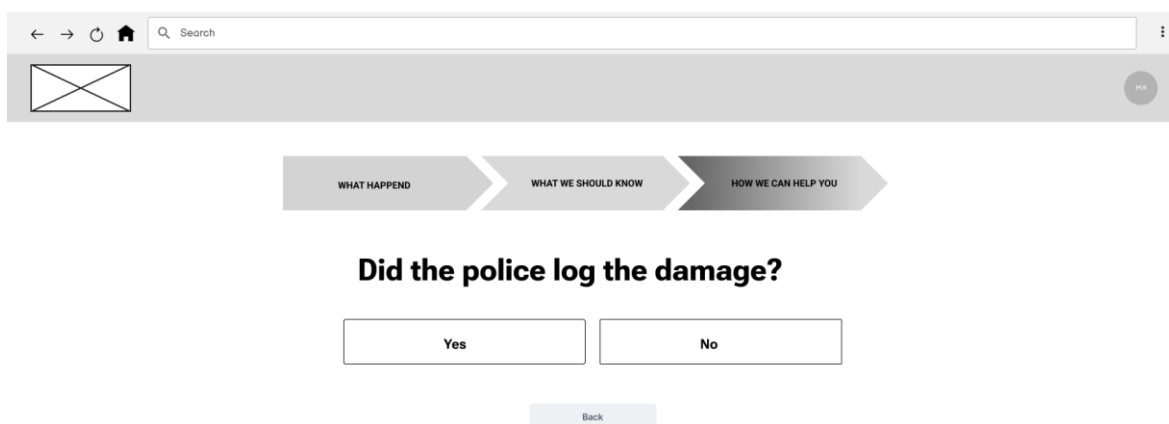
Add witness

Obrázek 24: Svědci – rozšířeno [Vlastní zpracování]

5.10 Policejní protokol

Během procesu vyplňování formuláře je uživatel také dotázán, zda o celém incidentu vede policie nějaký záznam. Pokud je odpověď kladná a policie má záznam, uživatel je požádán, aby vyplnil číslo spisu, které mu bylo uděleno. Toto číslo spisu bude pojišťovně sloužit k získání dalších informací o incidentu, které jsou zaznamenány v policejním protokolu.

Pokud však policie o incidentu neví, uživatel je vyzván, aby tuto událost nahlásil na policii. I přesto je možné pokračovat ve vyplňování formuláře hlášení škody a celý incident nahlásit policii dodatečně. Uživatel bude mít možnost odeslat potřebné informace o incidentu na policii poté, co dokončí proces vyplňování formuláře pro pojišťovnu.



The image shows a web browser interface with a search bar and navigation icons. Below the browser is a grey header bar with a placeholder icon on the left and a 'WV' logo on the right. The main content area features a horizontal flow of three grey arrow-shaped steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The third step is highlighted. Below the steps is the question 'Did the police log the damage?' in bold. Underneath the question are two rectangular buttons labeled 'Yes' and 'No'. At the bottom center is a 'Back' button.

Obrázek 25: Zaevidovala policie incident? [Vlastní zpracování]

← → ↻ 🏠 Search

WHAT HAPPEND WHAT WE SHOULD KNOW HOW WE CAN HELP YOU

Did the police log the damage?

Can you provide further information about this?

Police file number

Obrázek 26: Zaevidovala policie incident? – Ano [Vlastní zpracování]

← → ↻ 🏠 Search

WHAT HAPPEND WHAT WE SHOULD KNOW HOW WE CAN HELP YOU

Did the police log the damage?

Please report the incident to the police afterwards.

Of course, you can still carry out this damage report.

Obrázek 27: Zaevidovala policie incident? – Ne [Vlastní zpracování]

5.11 Platba

Jedná se o jednu z největších oblastí celého návrhu. V této sekci jsou zahrnuty všechny informace k tomu, jak má být vyplacena náhrada od pojišťovny.

Prvním krokem v této sekci je určení stavu opravy poškozeného vozidla. Pokud bylo vozidlo již opraveno, uživatel je vyzván k poskytnutí informací o účtu, na který má být částka vyplacena. V opačném případě, pokud vozidlo ještě nebylo opraveno, uživatel má možnost vybrat mezi několika možnostmi.

5.11.1 Vozidlo opraveno

V této části je důležité zjistit, zda bylo vozidlo již opraveno v době, kdy byla škoda hlášena. Pokud je odpověď kladná, což znamená, že vozidlo již prošlo opravou, proces se zjednodušuje. Uživatel je pouze vyzván k vyplnění informací o účtu, na který má být částka za provedenou opravu vozidla vyplacena.

The image shows a horizontal flow diagram with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. Below it is a form with the question 'Has the vehicle damage described already been repaired?'. There are two buttons: 'Yes' and 'No'. Below these buttons is a 'Back' button.

Obrázek 28: Platba: Bylo vozidlo již opraveno? [Vlastní zpracování]

The vehicle damage described already been repaired.

The image shows a form titled 'Who should we transfer the compensation to?'. It has two input fields: 'Account number:' with a sub-label 'Name and surname', and 'IBAN'.

Obrázek 29: Platba: Bylo vozidlo již opraveno? – Ano [Vlastní zpracování]

Z této stránky se přechází přímo na stránku souhrnu.

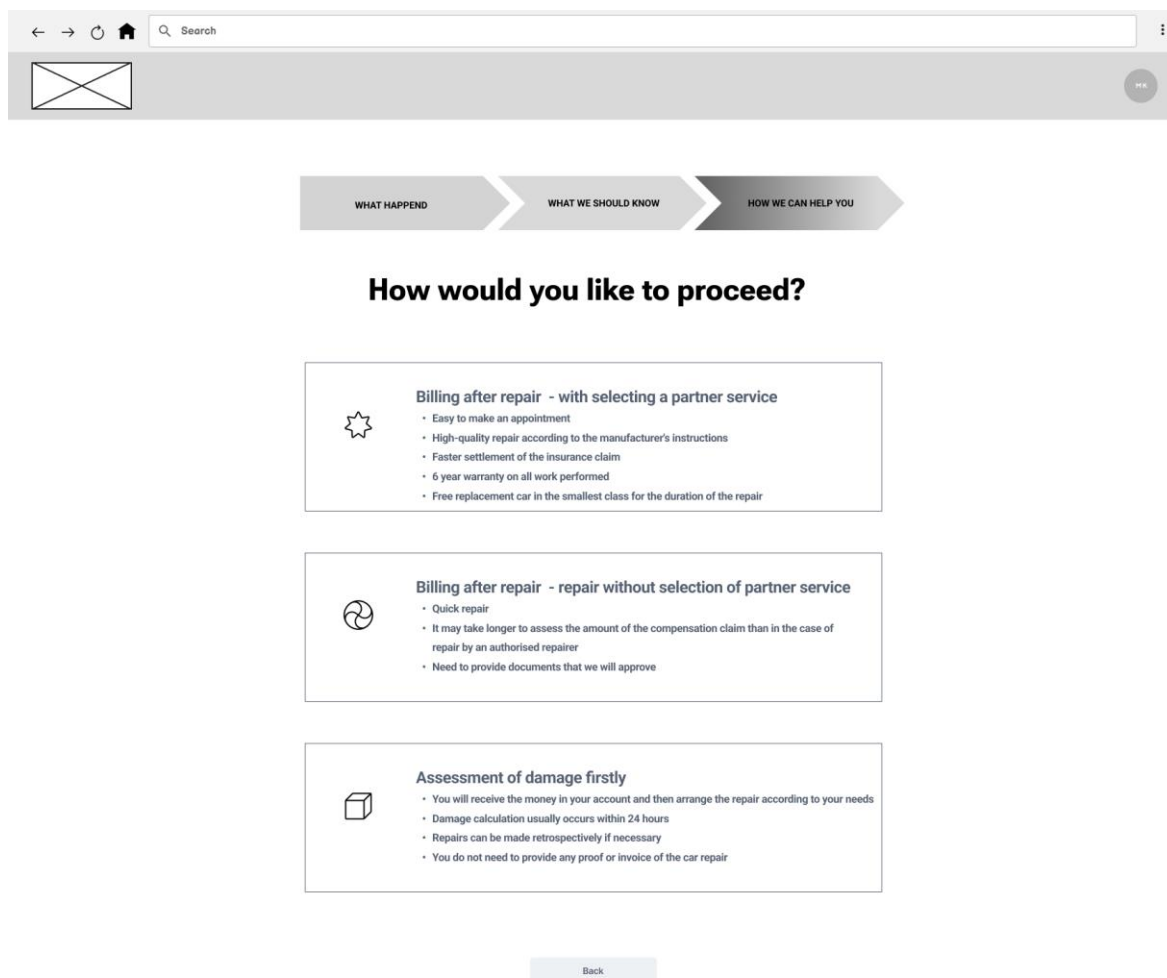
5.11.2 Vozidlo neopraveno – možnosti

Pokud vozidlo není v době, kdy je škoda hlášena, ještě opraveno, uživatel má k dispozici několik možností, jak postupovat dál:

- Platba po opravě – autorizovaný servis
- Platba po opravě – bez autorizovaného servisu
- Nejdříve posouzení škody

U každé z možností jsou uživateli stručně vysvětleny výhody a nevýhody dané volby, aby mohl lépe rozhodnout, který postup je pro něj nejvhodnější.

Po rozkliknutí vybrané možnosti je uživatel přesměrován na příslušnou stránku, kde může pokračovat v určování, jak škodu vyplatit.



Obrázek 30: Platba: Vozidlo neopraveno – Možnosti [Vlastní zpracování]

5.11.3 Platba po opravě – autorizovaný servis

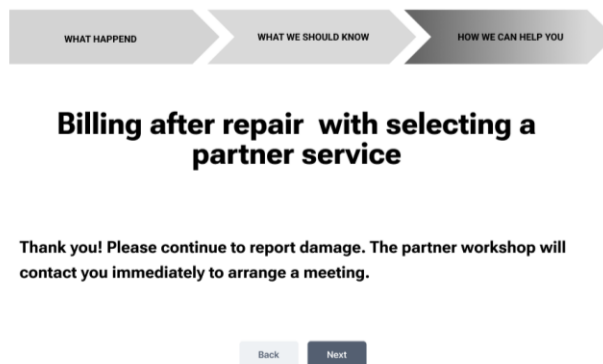
Pokud uživatel zvolí tuto možnost, je potřeba si vybrat autoservis, který je autorizován pojišťovnou. K tomuto účelu slouží interaktivní mapa, na které jsou zobrazeny nejbližší autorizované autoservisy.

Uživatel má možnost vyhledat tato servisní místa na základě poštovního směrovacího čísla, což umožňuje zobrazení autoservisů v dané lokalitě.

The screenshot shows a web browser window with a search bar containing '04109'. Below the search bar is a navigation bar with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The main heading is 'Billing after repair with selecting a partner service'. Below this, there is a prompt: 'Please enter your postcode to find partner workshops in your desired area.:' followed by a text input field containing '04109'. Another prompt reads: 'Please select a specific service from the map:'. Below this is a map showing several workshop locations. A callout box over the map highlights 'J&T SERVICE - Bermt, STV Strasse 2665, 564 51 Bermt'. To the right of the map are three buttons labeled 'Workshop 1', 'Workshop 2', and 'Workshop 2'. Below the map, the text reads: 'Selected service: J&T SERVICE - Bermt, STV Strasse 2665, 564 51 Bermt' and 'Contact for arrangements: 00 41 36 669 69 87; insurcrepair@jtservice.com'. At the bottom of the page are 'Back' and 'Next' buttons.

Obrázek 31: Platba po opravě – autorizovaný servis [Vlastní zpracování]

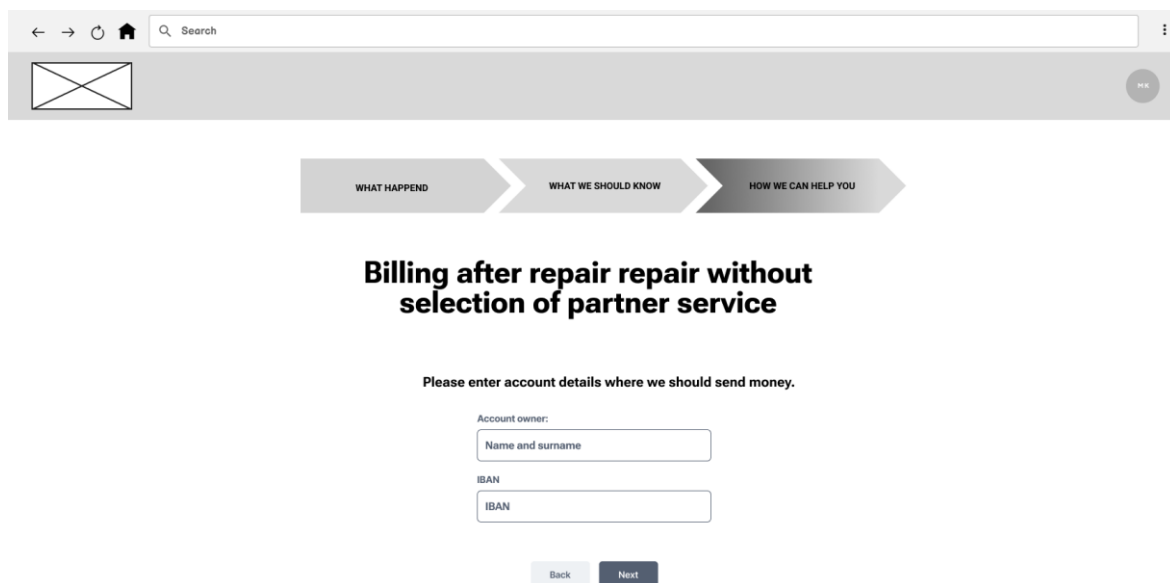
Po výběru konkrétního autoservisu z mapy se uživateli zobrazí pouze oznámení, jelikož autorizovaný autoservis bude kontaktován přímo pojišťovnou.



Obrázek 32: Platba po opravě – autorizovaný servis dokončení [Vlastní zpracování]
Po dokončení tohoto kroku uživatel přechází přímo na stránku s souhrnem informací, kde může ještě jednou zkontrolovat veškeré zadané údaje.

5.11.4 Platba po opravě – bez autorizovaného servisu

V případě volby možnosti "Platba po opravě – bez autorizovaného servisu" pojišťovna nemá žádnou předem dohodnutou spolupráci se servisním místem, kde by mělo být vozidlo opraveno. To znamená, že uživatel má volnost výběru servisu pro opravu svého vozidla. Po dokončení opravy je náhrada škody vyplacena přímo na účet vybraný uživatelem.



Obrázek 33: Platba po opravě – bez autorizovaného servisu [Vlastní zpracování]

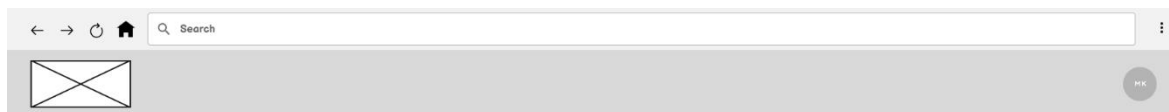
5.11.5 Nejdříve posouzení škody

Při této možnosti uživatel vyžaduje peníze od pojišťovny ještě před opravou vozidla. Jelikož ale pojišťovna potřebuje ověřit a posoudit rozsah škody, je uživatel požádán, aby vybral místo, kde k tomuto posouzení dojde. Uživatel má na výběr mezi dvěma možnostmi:

1. Autorizovaný servis: V tomto případě je posouzení škody prováděno autorizovaným servisem, který má s pojišťovnou dohodu o poskytování této služby. Technik servisu provede odhad škody a pojišťovna na základě tohoto odhadu vyplatí náhradu uživateli.
2. Určení místa pro posouzení: Uživatel má také možnost určit místo, kam má dojet technik pojišťovny a provést posouzení škody přímo na místě. To může být například místo, kde se incident stal, nebo jiné místo dle uživatelova uvážení.

The screenshot shows a web browser interface. At the top, there is a navigation bar with a search bar and a home icon. Below the navigation bar, there is a progress bar with three steps: 'WHAT HAPPEND', 'WHAT WE SHOULD KNOW', and 'HOW WE CAN HELP YOU'. The current step is 'Assessment of damage firstly'. Below this, there is a question: 'Where would you like the vehicle to be inspected?' and two radio button options: 'At one of our partner service centres.' and 'Inspection at a selected location by our technician'. A 'Back' button is visible at the bottom.

Obrázek 34: Nejdříve posouzení škody – možnosti [Vlastní zpracování]



Assessment of damage firstly

To calculate the damage, the vehicle must be inspected

Inspection at one of our partner service centres:

Please select a specific service from the map:



Selected service: J&T SERVICE - Berrnt, STV Strasse 2665, 564 51 Berrnt
Contact for arrangements: 00 41 36 669 69 87; insuncerepair@jtservice.com

The partner workshop will contact you immediately to arrange a meeting.



Obrázek 35: Nejdříve posouzení škody – autorizovaný servis [Vlastní zpracování]



Assessment of damage firstly

To calculate the damage, the vehicle must be inspected

Inspection at a selected location by our technician

Please fill in the location where you wish to have your vehicle inspected

Address

<input type="text" value="Street"/>	<input type="text" value="House number"/>
<input type="text" value="Town / City"/>	<input type="text" value="Post code"/>

Please fill out the contact information (phone or email) for the person who will allow our technician to inspect the vehicle:

<input type="text" value="Mobile"/>
<input type="text" value="Email address"/>

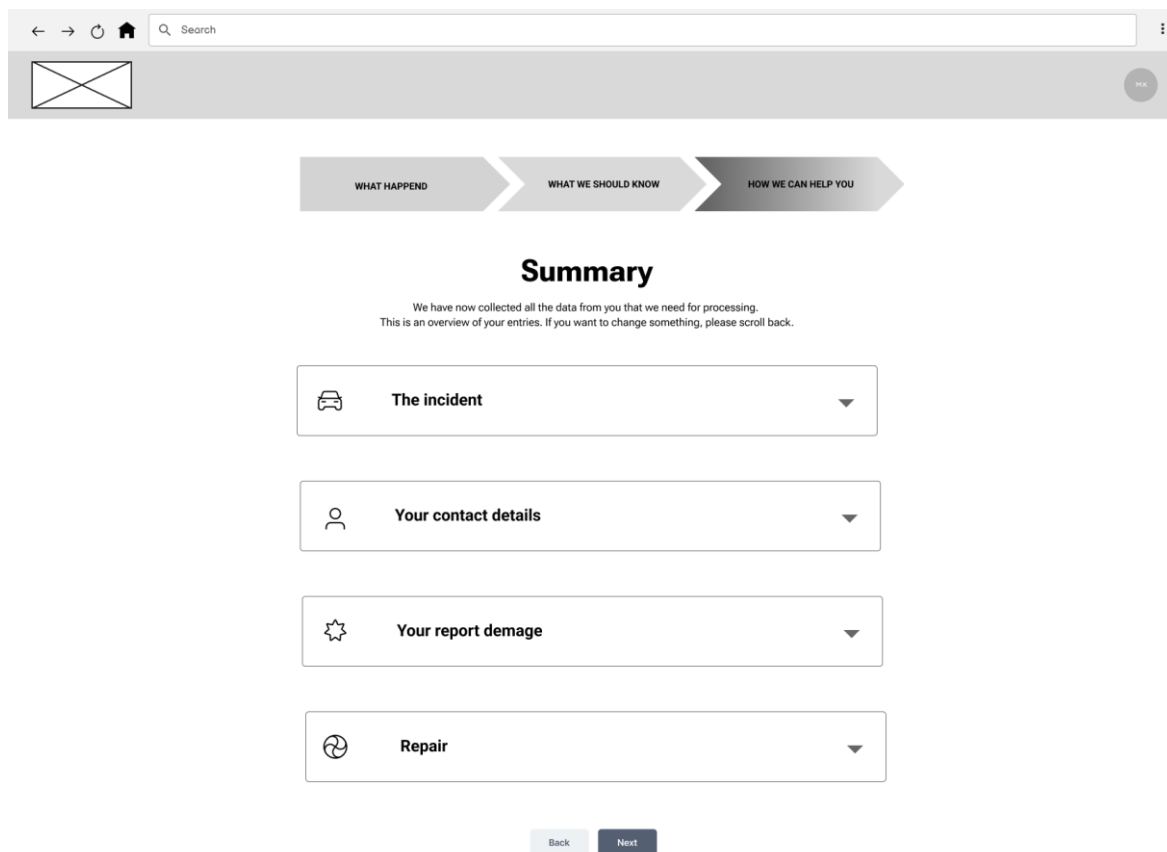
Obrázek 36: Nejdříve posouzení škody – určení lokace [Vlastní zpracování]

Jedná se o poslední krok výběru. Jakmile uživatel pokračuje z této stránky, dostává se na stránky souhrnu, kde zkontroluje zadané údaje před odesláním hlášení.

5.12 Souhrn

V souhrnu má uživatel možnost zkontrolovat všechny informace, které byly během vyplňování dotazníku zadány.

Zahrnuje to veškeré údaje o události, informace o vozidle, škodě, případně svědků nebo i policejním protokolu.



Obrázek 37: Souhrn [Vlastní zpracování]

Po zkontrolování informací má uživatel také možnost zanechat svou emailovou adresu pro odesílání informací o postupu vyřizování jeho požadavku. Tímto způsobem může uživatel sledovat stav svého požadavku a být průběžně informován o dalším postupu.

Obrázek 38: Souhrn – kontaktní email [Vlastní zpracování]

Na závěr je uživateli zobrazena informační stránka, která detailně popisuje, jak bude pojišťovna postupovat při vyřizování jeho požadavku. Tato stránka poskytuje uživateli jasný přehled toho, co může očekávat v dalších krocích a jak dlouho může celý proces trvat.

K dispozici je také tlačítko umožňující uživateli snadný návrat na hlavní stránku aplikace, kde může v případě potřeby zadat další žádost nebo provádět další činnosti.

Obrázek 39: Souhrn – poděkování [Vlastní zpracování]

6 Implementace

6.1 Vývoj

Po navrhnutí designu a průchodu celou aplikací následovalo samotné implementování aplikace.

Z důvodů komplexnosti celé práce, a především její variabilitě, nejsou veškeré portlety plně dodělány. Finální vzhled bude záležet na každém zákazníkovi (pojišťovně) a bude pravděpodobně vyžadovat úpravy.

6.1.1 Stažení Docker Image

Prvním krokem ve vývoji celé aplikace bylo stažení a instalace samotného Dockeru a vývojových prostředí.

Poté následovalo stažení a spuštění předpřipraveného Docker image, který je ve firmě k dispozici. Tento Docker image je jednotný napříč celým projektem a obsahuje základní Liferay instanci.

Po úspěšném spuštění a ověření správné funkcionality se mohlo přejít k dalšímu kroku.

6.1.2 Vývoj jednotlivých portletů

Vývoj jednotlivých portletů se téměř nelišil. Prvním krokem bylo vytvořit Typescriptový soubor a připravit základní strukturu.

Dále je potřeba doplnit HTML strukturu daného portletu a přidat funkčnost prvkům jako jsou tlačítka. V následujícím příkladu se jedná o portlet „Kdy k události došlo“. Zde je nutné přidat funkčnost poli pro vkládání dat – kalendáři. Tomuto prvku je třeba dodat výchozí hodnotu, kterou je dnešní datum.

```

1 import { Component, Inject, OnInit } from '@angular/core';
2 import { FormBuilder, FormGroup, Validators } from '@angular/forms';
3
4
5 @Component({
6   selector: "where",
7   template: `
8     <form [formGroup]="form">
9       <h1>Were there witnesses?</h1>
10      <section id="formBody">
11        <section class="centered">
12          <button class="modal2 modal2-white">Yes</button>
13          <button class="modal2 modal2-white">No</button>
14        </section>
15      </section>
16    </form>
17  `
18 })
19 export class WitnessesComponent implements OnInit {
20   form: FormGroup;
21
22   constructor(
23     @Inject(FormBuilder) public formBuilder: FormBuilder
24   ) {
25   }
26   ngOnInit(): void {
27     this.form = this.formBuilder.group({ date: ["DD.MM.YYYY"] });
28   }
29 }
30

```

Obrázek 40: Angular – Witnesses Component [Vlastní zpracování]

6.1.3 Stylování

Stylování pro Liferay probíhá mírně odlišně než při standardní tvorbě webů. Běžně se přikládají CSS soubory, které jsou přímo propojeny s danými html stránkami. Nebo jako v tomto případě při využití Angularu se vytvářejí komponenty, které obsahují HTML soubor a CSS soubor pro každou komponentu.

V Liferay ale toto stylování funguje mírně odlišněji. Celý proces stylování, ať už portletů samotných, nebo celé stránky, funguje pomocí takzvaných témat (theme). Kdy každá stránka využívá toto téma, které obsahuje kompletní CSS pro celou webovou aplikaci.

Vzhledem ke komplikovanosti samotného Liferaye jsou změny v samotném CSS poměrně komplikované. Proto byl využit tzv. liferay-generator, který je součástí pluginu Yeoman popisovaného v kapitole [3.5 Yeoman](#).

6.1.3.1 Stylování jednotlivých portletů

Po vytvoření struktury daného portletu je potřeba vytvořit jeho konkrétní vzhled pomocí CSS. Pokud je správně nastavená tvorba celé HTML a CSS struktury, je možné využít tyto styly opakovaně. Proto byly v celém projektu nastýlovány především unikátní portlety a poté znovu využity dané styly.

Pro stylování byl využit SASS což je obdoba CSS, ve které lze dané styly napsat. Na rozdíl od čistého CSS má několik funkcí, které zjednodušují stylování. Styly napsané v SASS jsou automaticky překompilovány do čistého CSS čímž je zaručena plná kompatibilita.

Pro oddělení byly veškeré CSS styly napsány do podtřídy `_custom`, které se načítá jako poslední, a tudíž přepisuje veškeré výchozí hodnoty od liferay témat.

6.1.3.2 Stylovaná samotné stránky

Mimo jednotlivých portletů je také potřeba nastýlovat samotnou stránku. Zde se jedná především o menu a patičku celého projektu. Celá aplikace je pouze demo verzí samotného produktu a je potřeba vzhled upravovat na základě koncového zákazníka. Z tohoto důvodu a také vzhledem k problémům popsaným v kapitole [6.2.3 Stylování stránky – Liferay theme](#) byt tento krok upozaděn a v aktuální fázi je přeskočen.

6.1.4 Přidání portletů do pořadí vyplňování

Po vytvoření samotného portletu ho bylo potřeba přidat do pořadí vyplňování formuláře. Jelikož každá z variant vyžaduje jinou sekvenci portletů, je pro každou variantu vytvořena separátní služba (service), která uchovává pořadí portletů.

```

1 import { Injectable } from '@angular/core';
2
3 import { WhereComponent } from '../components/where.component';
4 import { multiple_whatCausedDamageComponent } from '../components/MultipleOptions/multiple_whatCausedDamage.component';
5 import { WhenComponent } from '../components/when.component';
6 import { multiple_whatHasBeenDamagedComponent } from '../components/MultipleOptions/multiple_whatHasBeenDamaged.component';
7 import { DamageDescriptComponent } from '../components/damageDescript.component';
8 import { VehicleInfoComponent } from '../components/vehicleInfo.component';
9 import { AdditionalDocumentsComponent } from '../components/additionalDocuments.component';
10
11 @Injectable({ providedIn: 'root', })
12 export class EnviromentFlowService {
13
14     getFlow(){
15         return [
16             {
17                 component: multiple_whatCausedDamageComponent
18             },
19             {
20                 component: WhenComponent
21             },
22             {
23                 component: WhereComponent
24             },
25             {
26                 component: multiple_whatHasBeenDamagedComponent
27             },
28             {
29                 component: DamageDescriptComponent
30             },
31             {
32                 component: VehicleInfoComponent
33             },
34             {
35                 component: AdditionalDocumentsComponent
36             }
37         ] as {component: any, inputs: Record<string, unknown>}[];
38     }
39 }

```

Obrázek 41: Environmental Flow Service [Vlastní zpracování]

6.1.5 Sběr dat

Veškerá data, která jsou ve formuláři vyplněna, se ukládají do služby podle aktuální varianty. Dané služby jsou podobné jako soubor popsáný v [6.1.4 Přidání portletů do pořadí vyplňování](#).

Jediným rozdílem je vnitřní struktura souboru kde místo definování jednotlivých komponent a jejich pořadí se nachází jednotlivé datové typy.

6.2 Problémy

V této sekci jsou vypsány největší problémy, které při vývoji aplikace nastaly.

6.2.1 Požadavky na hardware

První problém nastal ještě před samotným vývojem aplikace. Pro rozběhnutí samotného Docker kontejneru s instancí Liferay je potřeba poměrně velké množství RAM. Stanice, na které byla aplikace vyvíjena měla 8 GB RAM a při spuštění daný kontejner plně nenaběhl, ale začal se opakovaně sám vypínat a znovu spouštěl.

Pokusem o přesun celého vývoje z operačního systému Windows 10 na Linux Mint, který je méně náročný na RAM, bylo zjištěno, že ani takováto úspora (přibližně 2 GB) nebyla dostačující. Proto byl nakonec na vývoj využit firemní notebook s 32 GB RAM.

Celý kontejner s aplikací při provozu aktuálně využívá přibližně 15 GB RAM (pouze samotný kontejner bez OS a vývojových prostředí). Z dalšího testování vyplynulo, že pro běžný vývoj (spuštěný prohlížeč s několika okny, vývojové prostředí, ...) je potřeba i 23 GB RAM.

Tudíž původní snahy o spuštění nemohli být úspěšné vzhledem k tomu, že celá stanice měla k dispozici celkem 8 GB.

6.2.2 Nasazování – Sdílené balíčky

Při jakékoli změně, která byla při vývoji provedena, se daná změna projeví až po zbudování (build) celého projektu. Při reálném vývoji každé nasazení trvá přibližně 20 sekund. Což sice není dlouhá doba, ale vzhledem k četnosti této operace je vhodné ji co nejvíce zkrátit. Proto vznikla snaha i při této implementaci čas zkrátit pomocí návodu přímo od Liferay Community [13].

Zkrácení probíhá tak, že jsou přibalované knihovny vyjmuty ze samotného projektu a zabaleny do Sdíleného balíčku, který není znovu sestavován při každém nasazení

aplikace. Tento sdílený balíček je sestaven jen jednou při vytvoření, nebo při případné změně a je na něj v projektu pouze okazováno.

Při implementaci tohoto řešení se ovšem neustále ukazovala chyba naznačující chybějící některé části balíčků. I po opakovaném ověření že je daná knihovna přibalena do sdíleného balíčku nebyla chyba odstraněna.

Vzhledem k tomu, že se jedná pouze o zjednodušení vývoje a nebylo nutné k vývoji samotnému, tak zprovoznění této funkcionality bylo upozaděno.

6.2.3 Stylování stránky – Liferay theme

Při využití liferay-generatoru je vygenerováno nové téma, které lze při tvorbě webové aplikace nasadit. Při vygenerování tématu pomocí liferay-generátoru vznikne několik souborů, ve kterých je možné psát změny, které se projevují ve výsledném tématu.

Prázdné téma i tak obsahuje například soubor `_custom`, do kterého je možné dělat tyto úpravy. Mimo soubor `_custom` by se zde měla vygenerovat i složka s vzorovými tématy. Tyto soubory by měl vývojář po výběru požadovaného tématu k souboru `_custom` a následně upravovat.

V této práci se při generování pomocí liferay-generátoru nepodařilo vygenerovat tato vzorová témata s jejich soubory. Proto byly veškeré úpravy vzhledu portletů psány do prázdného tématu.

Tímto se dali jednoduše nastylovat vlastní portlety, ale stylování stránky samotné by bylo značně komplikované vzhledem k velikosti a komplikovanosti Liferaye.

7 Shrnutí a diskuse výsledků

Po celou dobu tvorby práce probíhala diskuze se zaměstnanci firmy Cleverbee. Díky této diskuzi byly jednotlivé portlety mnohokrát upraveny, aby byly vizuálně jednotné a obsahovaly pouze informace nutné v každém kroku vyplňování formuláře. Po vytvoření aktuální verze projektu byl celý projekt firmou znovu překontrolován. Při této kontrole vzniklo od firmy následující vyhodnocení:

Tato bakalářská práce, realizovaná v rámci projektu Cleeo společnosti Cleverbee, se zaměřuje na digitalizaci procesů v pojišťovnictví s důrazem na uživatelskou přívětivost a efektivitu. Zvláštní pozornost z naší strany byla věnována "User Experience" (UX), což je klíčové pro zajištění, že konečný produkt je intuitivní, bezpečný a efektivní.

V rámci zadání byla klíčová rychlost procesu hlášení škod a její vliv na bezpečnost systému. Práce efektivně zpracovává tuto specifikaci s cílem minimalizovat možnost pro uživatele vytvářet nepravdivé scénáře a zneužívat systém. Tento aspekt byl podrobně zkoumán ve spolupráci s interními odborníky, kteří poskytli zpětnou vazbu pro optimalizaci UX designu.

Důraz na rychlé a efektivní zadávání informací při hlášení škod nejen zlepšuje uživatelský zážitek, ale také zvyšuje bezpečnost a spolehlivost celého systému. Výsledný design účinně omezuje prostor pro manipulaci a zneužití, což je přímo v souladu s požadavky zadání.

Tato práce tedy přináší značný přínos k produktu Cleeo, zvyšuje jeho tržní atraktivitu a posiluje pozici společnosti Cleverbee v oblasti pojišťovnictví. Je to příklad, jak pečlivě zvážené UX může přinést nejen lepší uživatelský komfort, ale také zvýšit celkovou bezpečnost a efektivitu digitálních systémů.

Originál této zprávy se nachází v přílohách.

8 Závěry a doporučení

V této práci se podařilo analyzovat nejběžnější data potřebná pro hlášení škodné události. Také byl navržen vzorový design a optimální „User experience“ jednotlivých portletů. Dále byl navržen optimální průchod vyplňováním formuláře. Tento průchod minimalizuje množství dat, které musí uživatel vyplnit. Tím je minimalizována možnost zneužití hlášení těchto událostí. Nakonec byly implementovány některé části pro demonstraci využití výsledné aplikace.

Dalšími projekty, které by mohly navázali na tuto práci, mohou být projekty, které budou mít za úkol vyřešit popsané problémy. Jednalo by se například o prozkoumání problematiky sdílených balíčků. Podrobnějším popsáním jejich funkcí, obecným použitím a jejich výhodami. Dále také práce na téma Liferay. Tato práce by mohla nejen vyřešit problém se zmíněnými tématy. Také by mohla například podrobněji popsat co Liferay je a jak funguje. Případně porovnat toto řešení s dalšími podobnými aplikacemi a v návaznosti na problém s hardwarovými požadavky navrhnout alternativy.

V návaznosti na práci samotnou je zde potenciál v jejím rozšíření a pokračování ve vývoji, tj. obecně upravit celou aplikaci pro potřeby jednotlivých koncových zákazníků (pojišťovny). Dále lze aplikaci rozšířit její integrací do celého produktu Cleo a unifikací datových struktur s výsledným produktem.

9 Seznam použité literatury

- [1] *Introduction to What is Liferay?* Online. C2023. Dostupné z: <https://help.liferay.com/hc/en-us/articles/360018183831-Introduction-to-What-is-Liferay>. [cit. 2024-03-24].
- [2] Angular - What is Angular? Online. 2023, Aug 15 2023. Dostupné z: <https://angular.io/guide/what-is-angular> . [cit. 2024-03-24].
- [3] TypeScript: JavaScript With Syntax For Types. Online. C2012-2024. Dostupné z: <https://www.typescriptlang.org/> . [cit. 2024-03-24].
- [4] Angular - What is Angular? Online. 2023, Aug 15 2023. Dostupné z: <https://angular.io/guide/what-is-angular#components> . [cit. 2024-03-24].
- [5] Angular - What is Angular? Online. 2023, Aug 15 2023. Dostupné z: <https://angular.io/guide/what-is-angular#dynamic-properties-and-attributes> . [cit. 2024-03-24].
- [6] Angular - What is Angular? Online. 2023, Aug 15 2023. Dostupné z: <https://angular.io/guide/what-is-angular#conditional-rendering> . [cit. 2024-03-24].
- [7] TypeScript: JavaScript With Syntax For Types. Online. 2023, Aug 15 2023. Dostupné z: <https://angular.io/guide/what-is-angular#rendering-a-list> . [cit. 2024-03-24].
- [8] *What is a Container? | Docker.* Online. C2024. Dostupné z: <https://www.docker.com/resources/what-container/>. [cit. 2024-03-24].
- [9] What Is Containerization | IBM. Online. Dostupné z: <https://www.ibm.com/topics/containerization> . [cit. 2024-03-24].
- [10] Docker. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2022, 21.2.2022. Dostupné z: <https://cs.wikipedia.org/wiki/Docker> . [cit. 2024-03-24].
- [11] *Figma: The Collaborative Interface Design Tool.* Online. Dostupné z: <https://www.figma.com/>. [cit. 2024-03-24].
- [12] *The web's scaffolding tool for modern webapps | Yeoman.* Online. Dostupné z: <https://yeoman.io/>. [cit. 2024-03-24].
- [13] *How to use Shared Library Javascript with Angular.* Online. 2022. Dostupné z: <https://liferay.dev/blogs/-/blogs/how-to-use-shared-library-javascript-with-angular-liferay>. [cit. 2024-03-24].
- [14] Schadensmeldung. Online. Dostupné z: <https://schadenservice.ergo.de/claim/vehicle/who-fills>. [cit. 2023-11-04].

[15] Kfz-Schadenmeldung (kfz--schaden--melden-allianz-de.translate.google). Online. Dostupné z: https://kfz--schaden--melden-allianz-de.translate.google/?AZDforeignUVID=azdallianzde%7Catl%7CvisitID%7C8a4c51f5-347d-4967-9948-1ad3d5f5ad57&AZDforeignUVID=azdallianzde%7Catl%7CvisitorID%7Ca5b76003-4990-4e0c-a655-cb3b8e1b9671&_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=cs&_x_tr_pto=wapp&_x_tr_hist=true. [cit. 2023-11-04].

10 Seznam obrázků

Obrázek 1: Příklad komponenty [4]	7
Obrázek 2: Příklad navázání tlačítka na data [5]	7
Obrázek 3: Příklad zobrazení prvku za podmínky [6]	8
Obrázek 4: Příklad vykreslování seznamů [7]	8
Obrázek 5: Kontejnerizace [8]	9
Obrázek 6: Vzhled aplikace Figma [11]	11
Obrázek 7: Hlavní rozcestník [Vlastní zpracování]	12
Obrázek 8: Kdy k události došlo? [Vlastní zpracování]	21
Obrázek 9: Kdy k události došlo? [Vlastní zpracování]	22
Obrázek 10: Kde k události došlo? [Vlastní zpracování]	23
Obrázek 11: Kdo nehodu způsobil? [Vlastní zpracování]	24
Obrázek 12: Co bylo poškozeno? [Vlastní zpracování]	25
Obrázek 13: Kdo formulář vyplňuje? [Vlastní zpracování]	25
Obrázek 14: Kdo formulář vyplňuje? – rozšířeno [Vlastní zpracování]	26
Obrázek 15: Jak k nehodě došlo? [Vlastní zpracování]	27
Obrázek 16: Co bylo ukradeno? [Vlastní zpracování]	27
Obrázek 17: Informace o vozidle – Značka a model [Vlastní zpracování]	28
Obrázek 18: Informace o vozidle – VIN [Vlastní zpracování]	29
Obrázek 19: Informace o vozidle – HSN/TSN [Vlastní zpracování]	29
Obrázek 20: Kde bylo vozidlo poškozeno? – model [Vlastní zpracování]	30
Obrázek 21: Kde bylo vozidlo poškozeno? – textové pole [Vlastní zpracování]	30
Obrázek 22: Fotografie a dokumenty [Vlastní zpracování]	31
Obrázek 23: Svědci [Vlastní zpracování]	32
Obrázek 24: Svědci – rozšířeno [Vlastní zpracování]	32
Obrázek 25: Zaevidovala policie incident? [Vlastní zpracování]	33
Obrázek 26: Zaevidovala policie incident? – Ano [Vlastní zpracování]	34
Obrázek 27: Zaevidovala policie incident? – Ne [Vlastní zpracování]	34
Obrázek 28: Platba: Bylo vozidlo již opraveno? [Vlastní zpracování]	35
Obrázek 29: Platba: Bylo vozidlo již opraveno? – Ano [Vlastní zpracování]	35
Obrázek 30: Platba: Vozidlo neopraveno – Možnosti [Vlastní zpracování]	36

Obrázek 31: Platba po opravě – autorizovaný servis [Vlastní zpracování]	37
Obrázek 32: Platba po opravě – autorizovaný servis dokončení [Vlastní zpracování]	38
Obrázek 33: Platba po opravě – bez autorizovaného servisu [Vlastní zpracování]	38
Obrázek 34: Nejdříve posouzení škody – možnosti [Vlastní zpracování]	39
Obrázek 35: Nejdříve posouzení škody – autorizovaný servis [Vlastní zpracování]	40
Obrázek 36: Nejdříve posouzení škody – určení lokace [Vlastní zpracování]	40
Obrázek 37: Souhrn [Vlastní zpracování]	41
Obrázek 38: Souhrn – kontaktní email [Vlastní zpracování]	42
Obrázek 39: Souhrn – poděkování [Vlastní zpracování]	42
Obrázek 40: Angular – Witnesses Component [Vlastní zpracování]	44
Obrázek 41: Environmental Flow Service [Vlastní zpracování]	46

11 Přílohy

Příloha č.1: Zpětná vazba od firmy Cleverbee solutions s.r.o.

Příloha č.2: Návrh průchodu hlášením škodní události

Příloha č.3: Implementace webové aplikace

Originální dokument se nachází v příloze.

Zpětná vazba k Hlášení škodných událostí – Produkt Cleeo

Tato bakalářská práce, realizovaná v rámci projektu Cleeo společnosti Cleverbee solutions s.r.o., se zaměřuje na digitalizaci procesů v pojišťovnictví s důrazem na uživatelskou přívětivost a efektivitu. Zvláštní pozornost z naší strany byla při návrhu věnována "User Experience" (UX), což je klíčové pro zajištění, že konečný produkt je intuitivní, bezpečný a efektivní.

V rámci zadání byla klíčová rychlost procesu hlášení škod a její vliv na bezpečnost systému. Práce efektivně zpracovává tuto specifikaci s cílem minimalizovat možnost pro uživatele vytvářet nepravdivé scénáře a zneužívat systém. Tento aspekt byl podrobně zkoumán ve spolupráci s interními odborníky, kteří poskytli zpětnou vazbu pro optimalizaci UX designu.

Důraz na rychlé a efektivní zadávání informací při hlášení škod nejen zlepšuje uživatelský zážitek, ale také zvyšuje bezpečnost a spolehlivost celého systému. Výsledný design účinně omezuje prostor pro manipulaci a zneužití, což je přímo v souladu s požadavky zadání.

Tato práce tedy přináší značný přínos k produktu Cleeo, zvyšuje jeho tržní atraktivitu a posiluje pozici společnosti Cleverbee v oblasti pojišťovnictví. Je to příklad, jak pečlivě zvážené UX může přinést nejen lepší uživatelský komfort, ale také zvýšit celkovou bezpečnost a efektivitu digitálních systémů.


Irena Skotálková
Marketing Manager

Cleverbee solutions s.r.o. +420 226 227 000
Tůrkova 2319/5b IČ: 26424525
CZ-149 00 Praha 4 DIČ: CZ26424525

Příloha č.2: Návrh průchodu hlášením škodní události

K bakalářské práci je přiložen soubor, ve kterém se nachází jednotlivé portlety včetně grafického znázornění, jak se při vyplňování postupuje.

Příloha č.3: Implementace webové aplikace

K bakalářské práci je přiložen soubor, ve kterém se nachází zdrojový kód nově vytvořené webové aplikace. Po rozbalení je nutné spustit Docker kontejner pomocí „docker-compose up“. Tento příkaz spustí Liferay portál, do kterého je poté potřeba nahrát ony portlety a téma.

Zadání práce z IS (eVŠKP)



Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání bakalářské práce

Autor: Jiří Jand'ourek
Studium: I2100211
Studijní program: B1802 Aplikovaná informatika
Studijní obor: Aplikovaná informatika

Název bakalářské práce: **Tvorba webové aplikace pro hlášení škodních událostí**
Název bakalářské práce AJ: Web application development for claims report

Cíl, metody, literatura, předpoklady:

Cílem práce je navrhnout a implementovat webovou aplikaci pro hlášení škod se zaměřením na uživatelské rozhraní aplikace. Následně bude vyhodnocen vliv rozložení těchto prvků a jejich struktury na "User Experience" a možnosti jejich vlivu na eliminaci zneužití hlášení těchto událostí.

Osnova

1. Úvod do problematiky a business pohledu
2. Analýza a návrh technického řešení
3. Implementace aplikace
4. Vyhodnocení a diskuze
5. Závěry a doporučení
6. Literatura

Česká asociace pojišťoven - ŠKODNÍ UDÁLOSTI (cap.cz): <https://www.cap.cz/skodni-udalosti>

Kfz-Schadenmeldung (kfz--schaden--melden-allianz-de.translate.google): https://kfz--schaden--melden-allianz-de.translate.google/?AZDforeignUVID=azdallianzde%7Catl%7CvisitID%7C8a4c51f5-347d-4967-9948-1ad3d5f5ad57&AZDforeignUVID=azdallianzde%7Catl%7CvisitorID%7Ca5b76003-4990-4e0c-a655-cb3b8e1b9671&_x_tr_sl=auto&_x_tr_tl=en&_x_tr_hl=cs&_x_tr_pto=wapp&_x_tr_hist=true

Schadensmeldung (ergo.de): <https://schadenservice.ergo.de/claim/vehicle/who-fills>

DXP - Liferay Learn: <https://learn.liferay.com/en/w/dxp/index>

Learn Design & Design Basics | Figma: <https://www.figma.com/resource-library/design-basics/>

Angular - Introduction to the Angular docs: <https://angular.io/docs>

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: Ing. Martina Husáková, Ph.D.

Datum zadání závěrečné práce: 15.10.2021