



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Milan Truchan

ServIS – webový systém pro firmy zabývající se opravami bagrů

Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Pavel Ježek, Ph.D.

Studijní program: Informatika

Studijní obor: Programování a vývoj software

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Poděkování.

Název práce: ServIS – webový systém pro firmy zabývající se opravami bagrů

Autor: Milan Truchan

Katedra: Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: Mgr. Pavel Ježek, Ph.D., Katedra distribuovaných a spolehlivých systémů

Abstrakt: Cílem tejto práce bolo vytvoriť softvérové dielo pre malé firmy zaoberajúce sa zemnými a výkopovými prácami, opravou a predajom bagrov, ktoré nemajú prístup k vhodnému softvérovému riešeniu pre svoju činnosť.

Potreba a správanie funkcionalít bola prekonzultovaná s majiteľom jednej z týchto firiem.

Vzniknutý softvér predstavuje riešenie problému, je schopný zobrazíť ponuku (stroje, prídavné zariadenia) a umožňuje užívateľom dopyt (v podobe emailov) na tieto ponuky. V aplikácii tiež existuje aukcia, kde sa dražia opravené bagre. Užívatelia si tiež môžu v aplikácii vytvoriť účet. Vymenované funkcionality môžu využívať prihlásení aj neprihlásení užívateľia. Bežní prihlásení užívateľia nemusia vyplňovať informácie o sebe vo formulároch pri dopytovaní sa na ponuku. Prihlásení admini majú možnosť spravovať stránku. Tj. pridávať nové, editovať a mazať existujúce ponuky, odpovedať na správy atď.

Title: ServIS – a web system for companies dealing with excavator repairs
Kľúčová slova: Informačný systém C# .NET Blazor Server MySQL

Author: Milan Truchan

Department: Department of Distributed and Dependable Systems

Supervisor: Mgr. Pavel Ježek, Ph.D., Department of Distributed and Dependable Systems

Abstract: Abstract.

Keywords: Information system C# .NET Blazor Server MySQL

Obsah

Úvod	2
0.1 Požiadavky na softvér	2
0.2 Alternatívne riešenia	3
0.3 Zhrnutie cieľov	3
1 Analýza	4
1.1 Voľba technológií	4
1.2 Voľba databázy	5
1.2.1 Návrh databázy, UML	5
1.2.2 ORM	5
1.3 Aukcia- odpočet a vyhodnocovanie	5
1.4 Posielanie a prijímanie správ	5
Záver	6
A Přílohy	7
A.1 První příloha	7

Úvod

Predstavme si malú firmu. Napríklad takú, ktorá práve vznikla. Takáto firma zatiaľ nemá zákazníkov. Ľudia ju nepoznajú. Preto sa firma potrebuje nejakým spôsobom zviditeľniť, napr. prostredníctvom vlastnej stránky. No tvorba stránky môže byť pracná alebo finančne náročná. A keďže sa bavíme o (začínajúcej) malej firme, tak finančný aspekt hraje veľkú rolu. Preto som sa rozhodol vytvoriť softvér, ktorý by umožnil malým firmám prezentovať svoju ponuku strojov a služieb. A to za žiaden alebo minimálny poplatok. Čitateľ by mohol namietat, že v súčasnosti predsa existujú riešenia, ktoré by majiteľom firiem umožnili vytvoriť si vlastný web zdarma. Čitateľ má síce pravdu, no tieto riešenia majú v sebe háčik. Podrobnejšie sa na nich pozrieme neskôr v podkapitole Alternatívne riešenia (0.2). Aby sme videli a pochopili plusy a mínusy alternatívnych riešení a toho môjho, poďme sa najprv pozrieť čo vyžadujú majitelia takýchto firiem.

0.1 Požiadavky na softvér

Po konzultácii s majiteľom jednej z firiem, boli vyhotovené tieto požiadavky:

- **P1 Dostupnosť**

Softvér by mal byť jednoducho dostupný každému užívateľovi. Či už ide o bežného zákazníka alebo administrátora.

- **P2 Náklady**

Keďže ide o malé firmy, pri ktorých sa predpokladá nízky rozpočet, chceme, aby náklady spojené s tvorbou a vedením softvéru boli minimálne alebo v ideálnom prípade žiadne.

- **P3 Minimálna obsluha softvéru**

Systém by mal fungovať a starať sa o seba „sám“. Teda softvér by mal fungovať tak, aby pri ňom nemusel ustavične sedieť človek a obsluhovať ho. Pracovníci firmy, vrátane majiteľa, majú svoju prácu a najímanie si nového pracovníka, ktorý by softvér obsluhoval, nie je z finančných dôvodov žiadúce.

- **P4 Predstavenie ponuky zákazníkom**

Systém by mal byť schopný prezentovať ponuku bagrov a prídavných zariadení zákazníkom.

- **P5 Aukcia**

Jednou z činností spomínaných firiem je oprava bagrov. Systém by mal byť schopný poskytnúť administrátorovi možnosť pridať opravený stroj do aukcie.

- **P6 Dopyt**

Bežnou praxou v tomto odvetví je, že cena strojov sa dopredu neudáva. Zákazník najprv vyjadrí záujem (pošle dopyt), prekonzultujú sa detaily medzi

potenciálnym kupcom a firmou, a až potom prebehne obchod. Z tohto dôvodu systém nebude fungovať na princípe ako bežné internetové obchody (tým myslím pridávanie do košíka s následnou platbou), ale bude fungovať na princípe posielania správ (dopytov). Takže systém by mal umožniť zákazníkovi poslať dopyt na položky (stroje, prídavné zariadenia), o ktoré majú záujem.

- **P7 Prístup k súčiastkam strojov**

Systém by mal umožniť administrátorovi jednoducho zistiť, aké súčiastky obsahuje konkrétny stroj.

- **P8 Registrácia a prihlásenie užívateľov**

Systém by mal umožniť bežným užívateľom možnosť registrácie a prihlásenia sa do systému. Po prihlásení získajú bežní užívatelia výhodu v tom, že do formulárov už nebudú musieť zadávať svoje údaje.

0.2 Alternatívne riešenia

Teraz, keď už vieme aké sú požiadavky, sa môžeme pozrieť na alternatívne riešenia a zhodnotiť plusy a mínusy.

Jedným z možných riešení by bolo použitie nejakého CMS systému (z ang. content management system), napr. WordPress. Autor práce síce nemá s platformou WordPress žiadne skúsenosti, ale po krátkom hľadaní na internete zistil, že pre túto platformu existuje aukčný plugin. S ním, by bolo dokonca možné na webe prevádzkovať i požadovanú aukciu. Ale toto riešenie by vyžadovalo znalosť platformy WordPress alebo by si majiteľ firmy musel najmúť niekoho, kto túto znalosť má. Keďže znalosť platformy nie je samozrejmosťou a najímanie si niekoho by bolo v rozpore s P2, túto alternatívu môžeme škrnúť.

Pre úplnosť ešte spomeniem, že jedným z riešení by bolo najmúť si inú firmu, ktorá by web vytvorila. No toto riešenie môže byť finančne náročné, a preto je taktiež v rozpore s P2.

0.3 Zhrnutie cieľov

Cieľom tejto práce je implementovať softvérový informačný systém určený pre firmy, ktoré sa zaoberajú predajom a opravou bagrov. Systém bude spĺňať požiadavky P1 až P7.

1. Analýza

V tejto kapitole sa zamyslíme nad tým, ako splniť požiadavky definované v Úvode (0.1).

Pre splnenie požiadavky P1 dáva veľmi dobrý zmysel vytvoriť naše riešenie ako webovú aplikáciu. Týmto spôsobom sa nemusíme starať o distribúciu programu k užívateľom. Stačí ak má zákazník (resp. admin) pripojenie na internet.

Je síce pravda, že voľba webovej aplikácie zahŕňa i voľbu hostingu. A ten nemusí byť lacný. To by mohlo byť v rozpore s P2. Ale je potrebné dodať, že ak by sme zvolili klasickú desktopovú aplikáciu, tak by sme ju museli nejakým spôsobom dodať zákazníkovi. A to by bolo nepraktické, prípadne by mohlo stáť takisto nejaké peniaze. Navyše práve webová aplikácia má potenciál pomôcť firme tak, že ju zákazník objaví pri surfovaní internetu.

Pre splnenie P4, P5 a P8 je jasné, že budeme potrebovať databázu. A to na to, aby si firmy vedeli samé tvoriť ponuku, ktorú si do databázy uložia. Po príchode zákazníka bude možné ponuku z databázy načítať a zobraziť. Podobne v prípade P8. Keď sa užívateľ zaregistruje, jeho údaje sa uložia v databáze a pri prihlásení sa z nej prečítajú a môžu použiť pre vyplnenie formulárov podľa potreby.

Znova sa vrátim k P5. Keďže ide o aukciu, budeme potrebovať nejaký mechanizmus, ktorý by vedel zabezpečiť odpočet, a takisto vyhodnotenie aukcie na pozadí. Taktiež si musíme rozmyslieť, ako sa má aukcia správať v rôznych situáciách.

Na to, aby sme splnili P6, musí byť náš softvér schopný posilať správy. Z podmienky P3 usudzujeme, že nikto nebude pri softvéri sedieť, a teda posielanie dopytov by nemalo mať povahu četu. Posielanie správ bude prebiehať prostredníctvom emailov. To nám vytvára novú požiadavku na softvér. Aby administrátor nemusel preklikať medzi svojou emailovou schránkou a naším systémom, bolo by dobre integrovať jeho schránku priamo do systému.

1.1 Voľba technológií

Po prejdení požiadaviek vieme, že chceme vytvoriť webovú aplikáciu s bohatým užívateľským rozhraním, ktorá by bola schopná posilať a prijímať správy, pracovať s databázou, umožnila nám autentikáciu a autorizáciu, a taktiež vykonávať prácu na pozadí. Pre túto úlohu sa hodia vysokoúrovňové jazyky, ako sú napríklad C# alebo Java. . . Na základe autorových skúseností si volíme jazyk C# a platformu .NET, ktorá je s ním spojená.

Platforma .NET nám pre vývoj webových aplikácií poskytuje framework ASP.NET alebo Blazor. Obe frameworky sú si podobné. Rozdiel nájdeme v tom, že Blazor umožňuje vytváranie komponent. Komponent si môžeme predstaviť ako logickú časť stránky (napr. tabuľka, tlačidlo. . .). Po zedefinovaní komponentu ho vieme „recyklovať“. Tým myslím to, že ho môžeme použiť na viacerých miestach na webe. Na každom mieste sa bude správať a vyzeráť rovnako (príp. vieme jeho správanie meniť pomocou parametrov). Táto myšlienka komponentov sa autorovi páči, dobre sa s ňou pracuje a neskôr si ukážeme ako nám pomôže vyriešiť problém s odpočtom.

Blazor poskytuje viacero hosting modelov. V čase rozhodovania existovali dva: Blazor WebAssembly a Blazor Server. Výber WebAssembly by zahŕňal niekoľko problémov. Pri prvotnej návšteve stránky sa musia klientovi stiahnuť zdrojové kódy aplikácie. To môže chvíľu trvať a mohlo by to odradiť nových potenciálnych zákazníkov. V prípade Blazor Server tento problém nemáme, pretože kód beží na serveri a užívateľovi sa servíruje už len prerenderovaný HTML, CSS, JavaScript kód stránky. Z rovnakého dôvodu sú weby vytvorené Blazor Serverom SEO-friendly (čo znamená, že sú dohľadateľné vyhľadávačmi, akým je napríklad Google). V prípade WebAssembly...

1.2 Voľba databázy

V predošlom texte sme spomenuli, že pre splnenie P4 a P5 budeme potrebovať databázu, ale akú?

1.2.1 Návrh databázy, UML

Obrázok, opis.

1.2.2 ORM

Prečo by sme chceli ORM.

1.3 Aukcia- odpočet a vyhodnocovanie

V tejto podkapitole predstavím BackgroundServices?, vysvetlím ako odpočítavať (len 1 timer, v osobitnej komponente kvôli rerenderom), ako funguje vyhodnocovanie- rozne scenare (co sa stane ak mame vitaza, co sa stane ak nemamee vitaza).

1.4 Posielanie a prijímanie správ

V tejto podkapitole rozoberieme spôsoby ako posilať/prijímať správy. Gmail api? AE.net? Mailkit?

Závěr

A. Přílohy

A.1 První příloha