

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Webové aplikace nad databází SAP

Bakalářská práce

Autor: Michal Ludvíček

Studijní obor: Informační management – kombinovaně

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.

Hradec Králové

Srpen 2022

Prohlášení:

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

V Hradci Králové dne 14.08.2022

Michal Ludvíček

Poděkování:

Tímto děkuji své vedoucí bakalářské práce Mgr. Daniele Ponce Ph.D. za cenné rady, podněty a připomínky při zpracování této bakalářské práce.

Anotace

Tato bakalářská práce se zabývá tvorbou webových aplikací nad databází SAP Business One. Teoretická část je zaměřená na seznámení s technologiemi použitými při vývoji aplikace v praktické části. Dále práce popisuje SAP komponenty potřebné pro její vývoj, který je demonstrován vytvořením tří aplikací pomocí frameworku SAPUI5. Aplikace je nasazena v prostředí SAP HANA. Po dokončení vývoje byly aplikace nasazeny do provozu. Práce může sloužit začínajícím vývojářům k seznámení s komponentami využitelnými během vývoje aplikací nad databází SAP Business One.

Klíčová slova

SAP Business One, SAPUI5, SAP HANA, Framework, aplikace

Annotation

The aim of the bachelor thesis is to present the process of creating web applications over the SAP Business One database. The theoretical part is focused on familiarizing with the technologies used in the development of the application in the practical part. Furthermore, the thesis describes the SAP components required for its development, which is demonstrated by creating three applications using the SAPUI5 framework. The application is deployed in the SAP HANA environment. After the development was completed, the applications were deployed into production. The work can serve novice developers to familiarize themselves with the components that can be used during the development of applications over the SAP Business One database.

Keywords

SAP Business One, SAPUI5, SAP HANA, Framework, application

Obsah

1	Úvod	10
2	Cíl práce, metodika	11
3	SAP Business One	12
3.1	<i>OData Protocol.....</i>	<i>12</i>
3.2	<i>SAP Fiori.....</i>	<i>13</i>
3.3	<i>SAP Fiori launchpad</i>	<i>13</i>
4	Framework pro uživatelské rozhraní SAP UI5	14
4.1	<i>Ukázka připojení k databázi.....</i>	<i>17</i>
4.2	<i>Data Binding</i>	<i>18</i>
5	Systém pro řízení databáze SAP HANA	21
6	Vrstva služby SAP (SAP Service Layer)	22
7	Ukázková aplikace nad SAP (analýza a návrh)	24
7.1	<i>Přehled montážních zakázek.....</i>	<i>24</i>
7.2	<i>Sledování práce.....</i>	<i>26</i>
7.3	<i>Příjem požadavků na servis.....</i>	<i>28</i>
8	Ukázková aplikace nad SAP (implementace)	31
8.1	<i>Přehled montážních zakázek.....</i>	<i>32</i>
8.2	<i>Sledování práce.....</i>	<i>33</i>
8.3	<i>Příjem požadavků na servis.....</i>	<i>34</i>
9	Dosažené výsledky	36
10	Shrnutí práce	37

11 Závěr a doporučení.....	38
12 Použitá literatura	39
13 Přílohy.....	41

Seznam obrázků

Obrázek 1: MVC architektura [autor].....	16
Obrázek 2: Ukázka vytvoření metody GET pomocí ajax [autor]	18
Obrázek 3: Ukázka vytvoření textového prvku [autor]	18
Obrázek 4: Ukázka vytvoření JSON modelu [autor]	19
Obrázek 5: Ukázka vytvoření modelu a přiřazení k jádru SAPUI5 [autor]	19
Obrázek 6: Ukázka použití modelu v jazyku xml [autor]	19
Obrázek 7: Ukázka vytvoření Resource Model [autor]	20
Obrázek 8: Ukázka obsahu souboru i18n.properties [autor]	20
Obrázek 9: Ukázka použití modelu v jazyku xml [autor]	20
Obrázek 10: Volání metody GET [autor: aplikace Postman].....	22
Obrázek 11: Výsledek metody GET [autor: aplikace Postman]	23
Obrázek 12: Filter bar pro tabulku [autor]	24
Obrázek 13: Tabulka s daty [autor]	24
Obrázek 14: Detail montážní zakázky [autor]	25
Obrázek 15: Pohled na časovou osu [autor].....	25
Obrázek 16: Use case diagram [autor]	26
Obrázek 17: Filtr pro data [autor].....	27
Obrázek 18: Tabulka s daty [autor]	27
Obrázek 19: Ganttův diagram [autor].....	28
Obrázek 20. Hlavní stránka [autor]	29
Obrázek 21: Napovídání dat [autor].....	29
Obrázek 22: Hledání podle města [autor]	30
Obrázek 23: Vytvoření servisního hlášení [autor].....	30

Seznam zkratek

ERP	System pro plánování podnikových zdrojů
API	Rozhraní pro programování aplikací
OData	Open Data Protocol
AJAX	Asynchronous JavaScript and XML

1 Úvod

Společnost SAP patří mezi jedny z nejvýznamnějších poskytovatelů systémů pro plánování podnikových zdrojů. Mezi její produkty patří například SAP Business One, který se soustředí na malé a střední podniky. Těm pomáhá s automatizací klíčových podnikových funkcí a umožňuje firmám nahradit několik stávajících programů jedním plně funkčním systémem. Tento efektivní systém dokáže automatizovaně řídit například oblast financí, účetnictví, skladů a distribucí, nebo řízení lidských zdrojů.

Tato bakalářská práce popisuje návrh a vývoj webových aplikací nad databází SAP. V této práci budou popsány aplikace Přehled montážních zakázek, Sledování práce a Příjem požadavků na servis. Přehled montážních zakázek slouží pro přehled zakázek firmy v období, které si filtruje uživatel dle potřeby. Aplikace sledování práce se využívá pro přehled plnění normy jednotlivých zaměstnanců firmy. Aplikace Příjem požadavků pro servis slouží k vytvoření tiketu servisního hlášení. Umožňuje obsluze aplikace rychlé dohledání informací o zákazníkovi přímo z databáze, vytvoření tiketu s popisem poruchy a z něho následné vytvoření servisního hlášení.

2 Cíl práce, metodika

Cílem bakalářské práce je představit postup tvorby webových aplikací nad databází SAP a demonstrovat jej ukázkovou aplikací. Řešení, které bude realizováno a popsáno v této práci, může sloužit i jako vzor pro začínajícího programátora. Seznámí ho s komponentami, používanými při vývoji aplikací nad databází SAP Business One. Celkem budou v této práci vytvořeny tři ukázkové responsivní aplikace. Tato práce se soustředí na vývoj frontendu webových aplikací s použitím frameworku SAPUI5.

Prvním krokem při vývoji aplikace je komunikace se zákazníkem a analýza jeho požadavků. Na základě toho se následně navrhne řešení využívající vhodné komponenty z frameworku SAPUI5. Například pro aplikaci Přehled montážních zakázek budou vybrány předem nastýlované prvky `sap.ui.comp.Filterbar` pro filtrování dat a pro zobrazení dat bude zvolen prvek `sap.m.Table`. Tyto prvky budou následně upraveny podle naší potřeby. Takto se bude postupovat i s ostatními prvky ve všech aplikacích.

3 SAP Business One

V dnešním technologicky vyspělém světě, kde je mnoho firem vzájemně propojených integrovanými systémy, pracuje většina firem na systémech ERP, mezi které patří i SAP Business One. SAP Business One je software prodáváný německou společností SAP SE, který slouží k řízení podniku, přesněji k plánování podnikových zdrojů. Určen je především pro malé a střední firmy. Mezi dodavatele tohoto software v České republice patří především firma Versino CZ, která je od roku 2009 oceněna jako SAP Gold partner. [1][2]

SAP Business One je zaměřen na automatizaci podnikových funkcí v oblasti výroby, účetnictví, financí, nákupu, zásob, prodeje a reklamací. Jsou v něm dostupné i různé reporty a analýzy a také díky němu lze mít přehled nad lidskými zdroji. Tento software přináší firmám mnoho výhod, mezi které patří především lepší sdílení informací v rámci firemní organizace a s ní související mnohem rychlejší reakci na dotazy a požadavky jejich zákazníků. [1][2]

3.1 OData Protocol

Jedním ze základních nástrojů, které SAP Business One využívá, je OData Protocol. Jedná se o standardní webový protokol, který se používá pro dotazování a aktualizaci dat ze systému SAP. Při jeho použití se využívají běžné webové technologie, jako například HTTP. Data jsou předávána pomocí rozhraní API. OData vytvoří odkaz na službu a k té poté může být přístupováno přes internet. Data následně mohou být vytvářena, editována, čtena i smazána. [3]

Dotazovací jazyk se zde používá přímo v URL. Jestliže tedy dojde ke změně URL adresy, změní se i data, která se pomocí OData posílají. Díky tomu lze získaná data kontrolovat a zákazníkům odesílat pouze tu část dat, ke které mají oprávnění přistupovat. [3]

3.2 SAP Fiori

SAP Fiori je nové designové prostředí pro softwarový program a aplikace. Jedná se o pravidla, jak mají aplikace vypadat, i jak se mají chovat, každý prvek má svoje místo. Programátorovi tedy usnadňují především návrhovou část. SAP Fiori definuje různé typy aplikací, jejich vzhled i chování. Poskytuje více než 300 aplikací založených na rolích. Patří mezi ně například lidské zdroje, výroba a financování. [4]

3.3 SAP Fiori launchpad

Jedná se o webovou aplikaci, která je založená na uživatelských rolích, což znamená, že v jedné organizaci mohou mít různí lidé různá oprávnění k jednotlivým aplikacím. Z jednoho místa nabízí přístup k mnoha SAP aplikacím nasazených na více platformách. Aplikace zde jsou prezentovány jako dlaždice. Aby uživatel Fiori launchpadu viděl jednotlivé dlaždice, musí k nim mít přidělené oprávnění. Na dlaždicích je možné zobrazovat krátké informace, například počet servisních hlášení, aniž by bylo nutné aplikaci přímo spouštět. Přístup do aplikací z Fiori launchpadu je možný například pomocí SAP GUI pro HTML. Aplikace se pak uživateli otevře na nové kartě nebo v novém okně prohlížeče. Jako další varianta je vložení SAPUI5 aplikace přímo do Fiori launchpadu. [5][6]

4 Framework pro uživatelské rozhraní SAP UI5

SAPUI5 je nejnovější programovací sada uživatelského rozhraní od společnosti SAP. Slouží k navrhování SAP Fiori aplikací, které se staly novým standardem, jak by měly moderní SAP aplikace vypadat. Aplikace, které se pomocí SAPUI vytvoří, fungují napříč prohlížeči a zařízeními, poskytují bohaté ovládací prvky uživatelského rozhraní a podporují rozvržení jazykových překladů. Před začínáním seznamování s jazykem SAPUI5 je vhodné umět základy jazyka JavaScript. [7]

Jedná se o nástroj na straně klienta, který se skládá z kaskádových stylů (CSS), HTML5 a Javascriptu. Díky tomu můžeme mít oddělený frontend od backendu. SAPUI5 je k dispozici pro zákazníky SAPu pomocí doplňku SAP NetWeaver. Je k dispozici ke stažení ze SAP Service Marketplace a na SAP Cloud Platform. SAP neustále pokračuje ve zlepšování frameworku a postupně vydává nové verze, aby se jazyk držel s trendy a novinkami ve světě. [7]

Knihovny nástrojů SAPUI5 k vytváření, testování a nasazení používají Node.js a jeho správce balíčků npm. Dostupné balíčky obsahují:

UI-cli

Tento balíček obsahuje rozhraní příkazového řádku, které může řídit všechny ostatní balíčky. Můžeme díky němu spustit sestavení, spustit webový server a inicializovat nástroje pro nový projekt.

UI5-server

Tento balíček obsahuje samostatný webový server pro spuštění místního projektu SAPUI5 během vývoje.

UI5-project

Tento balíček umožňuje aplikačnímu projektu správně sledovat všechny jeho závislosti.

UI5-Builder

Tento balíček umožňuje optimalizaci pro nasazení verzí webových projektů.

UI5-fs

Tento balíček spravuje interní reprezentaci struktury souborů projektu, což je užitečné pro dostupnost nástrojů napříč platformami a optimalizaci sestavení.

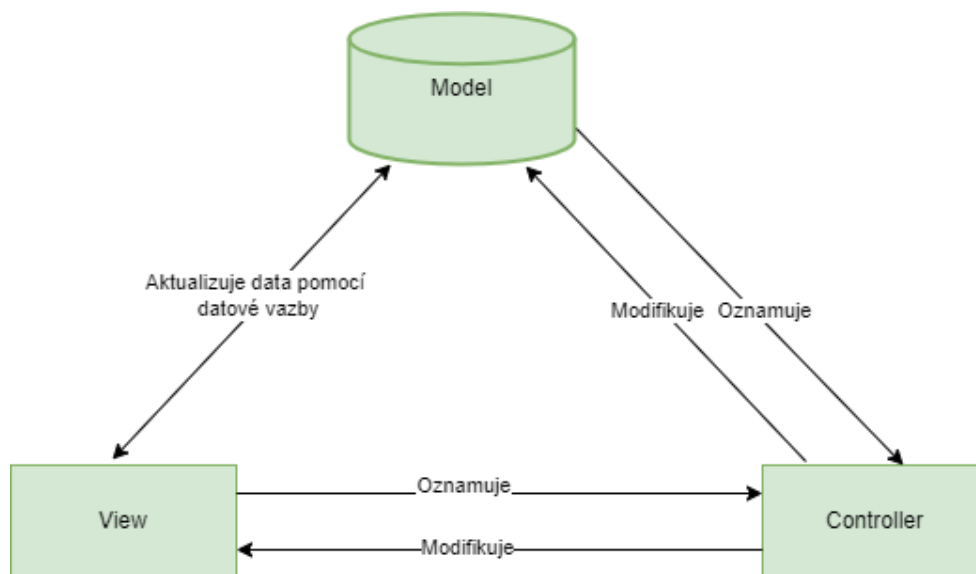
UI5-logger

Tento balíček se používá pro pokročilé protokolování sestavení a vývojové činnosti.
[7]

SAPUI5 umožňuje snadno přijímat data z jakékoli služby. To umožňuje kratší čas aktualizací a nejmodernější vývoj frontendu, zatímco funkce backendu zůstane nedotčená. SAPUI se skládá z několika knihoven, do kterých je organizováno jádro a ovládací prvky nástrojů. Ovládací prvky definují vzhled a chování částí obrazovky aplikace. Je zde přístupný velký počet ovládacích prvků, pomocí nich lze rychle tvořit uživatelské rozhraní a snadno připojit k backendové službě. Existuje zde více než 200 ovládacích prvků, od různých vstupních formulářů po listy a analytické tabulky, které umožňují seskupování, třídění a filtrování prvků. [7]

Model View Controller v SAPUI5

SAPUI se u vytváření aplikací řídí vzorem MVC. Je zde k dispozici několik typů modelů, které nám usnadňují připojení k backendovým službám a údržbu dat v aplikaci. Všechny ovládací prvky mají vazbu (binding) na model nebo zobrazení (view). K dispozici jsou různé typy zobrazení, které nám při vývoji umožňují zvolit, jestli chceme své zobrazení (view) psát v JavaScriptu, XML, JSON nebo HTML. Doporučené je psát v XML, kvůli snadné čitelnosti a udržovatelnosti. Dokumentace SAPUI uvádí příklady projektů také především v XML. Architektura je zobrazena na Obrázku 1.



Obrázek 1: MVC architektura [autor]

Jako další architekturu je možné zmínit například Model View Presenter, která z MVC vychází. Model má zde stejnou funkci, view zde také vykresluje uživatelské rozhraní a dále čeká na vstup od uživatele, který ho dále zasílá na presenter. Ten zde má funkci řídicí logiky a dohlíží na úspěšnost celého procesu. Tato architektura se v SAPUI5 nepoužívá. [8]

Kompatibilita napříč prohlížeči

SAPUI dokáže zjistit, jaké aktuálně uživatel používá zařízení, operační systém, prohlížeč a verzi prohlížeče. Zjišťuje také velikost obrazovky a podle toho se přizpůsobí. Upravuje velikost a polohu ovládacích prvků, když se na zařízení přepíná režim zobrazení na výšku nebo na šířku. Většina ovládacích prvků se buď přizpůsobí různým šířkám obrazovky, nebo je lze nakonfigurovat, aby vyhovovali různým situacím při používání aplikace. Mezi běžné možnosti konfigurace patří nastavení widths, margins, paddings, layouts a nastavení chování prvků při změně obrazovky, například které sloupce tabulky budou na menších zařízeních vidět a které budou skryté, nebo na mobilních zařízeních možnost seskupování prvků pod sebe. [7]

Lokalizace

Někteří zákazníci používají aplikace ve více zemích, a proto se musí přizpůsobit jazykovým a lokalizačním funkcím. SAPUI má mechanismus detekce národního prostředí. Většina ovládacích prvků, které vyžadují lokalizaci se přizpůsobí zjištěnému jazykovému nastavení a zobrazuje data, hodnoty a měny v daném formátu. Musí ovšem existovat překlad. Překlady lze uložit do souboru tomu určenému. V jedné aplikaci lze mít více takových to překladů. Jeden pro každý jazyk, ve kterém by měla být aplikace dostupná. [7]

4.1 Ukázka připojení k databázi

K připojení k databázi se používá především AJAX, který je složený z JavaScriptu a XML. Používá kombinaci vestavěného XMLHttpRequest objektu prohlížeče pro vyžádání dat z webového serveru a JavaScript s HTML pro zobrazení nebo použití dat. Aplikace mohou k přenosu dat používat xml, ale zároveň i formát JSON nebo prostý text. AJAX umožňuje asynchronní aktualizaci webových stránek. To znamená, že je možné aktualizovat části webové stránky, aniž bychom museli znovu načíst celou stránku. [9]

Průběh AJAX volání:

1. Na webové stránce vznikne událost, například při inicializaci stránky, nebo po stisku nějakého tlačítka.
2. Pomocí JavaScriptu se vytvoří objekt XMLHttpRequest.
3. Tento objekt odešle požadavek na webový server.
4. Server zpracuje požadavek.
5. Server odešle odpověď zpět na webovou stránku.
6. JavaScript přečte odpověď.
7. Následně se odpověď na stránce zpracuje podle potřeby, například se takto naplní nebo aktualizuje tabulka s daty. [9]

Ukázka kódu:

```
$.ajax({
  type: "GET",
  url: url,
  data: JSON.stringify(data),
  crossDomain: true,
  contentType: 'application/json',
  dataType: 'json',
  beforeSend: function (xhr) {
    xhr.setRequestHeader("B1SESSION", this.oStorage.get("B1SESSION"));
  }.bind(this),
  success: function (res) {

    this.getView().getModel("DataModel").setProperty("/TableData", res);

  }.bind(this),
  error: function (body) {

    MessageBox.error(body);

  }.bind(this)
});
```

Obrázek 2: Ukázka vytvoření metody GET pomocí ajax [autor]

4.2 Data Binding

Vytvoření textového prvku uživatelského rozhraní, který zobrazuje pevně daný textový řetězec (Obrázek 3).

```
new Text({text: "Hello, world!"}).placeAt("content");
```

Obrázek 3: Ukázka vytvoření textového prvku [autor]

Pro manipulaci s daty aplikace slouží model. Osobní data v modelu můžeme definovat pomocí několika formátů, například formát JSON nebo XML. Do modelu načteme data buď ze souboru uloženého lokálně u klienta, nebo vyžádáním dat z webového serveru. Model funguje na straně klienta a operace jako například třídění a filtrování dat se tedy provádí lokálně.

JSON model předává aplikaci text jako objekt. Aby měly všechny ovládací prvky v aplikaci přístup k modelu, je potřeba ho vytvořit a nastavit objekt modelu jádru SAPUI5 (Obrázek 4).

```
// Vytvoření JSON modelu
var oModel = new JSONModel({
    greetingText: "Hello, world!"
});

// Přiřazení objektu modelu k jádru SAPUI5
sap.ui.getCore().setModel(oModel);

new Text({text: "{/greetingText}"}).placeAt("content");
```

Obrázek 4: Ukázka vytvoření JSON modelu [autor]

Obecně lze říct, že objekt modelu by se však neměl vázat na jádro, ale na potřebný pohled. V tomto příkladě se jedná o jednostrannou vazbu, kde se data v aplikaci pouze zobrazí. Když bychom chtěli data v modelu z aplikace i upravovat, například v nějakém vstupním poli, je nutné použít obousměrnou vazbu dat.

```
// Vytvoření JSON modelu
var oModel = new JSONModel({
    firstName: "Michal",
    lastName: "Novák",
    enabled: true,
});

// Přiřazení objektu modelu k jádru SAPUI5
sap.ui.getCore().setModel(oModel);

// Zobrazení XML pohledu nazvaného "HomePage"
new XMLView({
    viewName: "sapui5.helloWorld.view.HomePage"
}).placeAt("content");
```

Obrázek 5: Ukázka vytvoření modelu a přiřazení k jádru SAPUI5 [autor]

```
<content>
    <Label text="First Name" class="sapUiSmallMargin"/>
    <Input value="{/firstName}" valueLiveUpdate="true" width="200px" />
    <Label text="Last Name" class="sapUiSmallMargin"/>
    <Input value="{/lastName}" valueLiveUpdate="true" width="200px" />
</content>
```

Obrázek 6: Ukázka použití modelu v jazyku xml [autor]

Resource Models

V některých aplikacích je nutné použití vícejazyčných překladů. K tomuto účelu slouží Resource Models neboli zdrojové modely (Obrázek 7). Veškerý text, který se bude zobrazovat v pohledech aplikace musí být umístěný ve speciálním souboru nazvaném i18n.properties (Obrázek 8).

```
// Vytvoření resource bundle pro texty specifické pro konkrétní jazyk
var oResourceModel = new ResourceModel({
    bundleName: "sap.ui.demo.db.i18n.i18n"
});

// Přiřazení objektu modelu k SAPUI5 jádru používající název "i18n"
sap.ui.getCore().setModel(oResourceModel, "i18n");
```

Obrázek 7: Ukázka vytvoření Resource Model [autor]

```
# Popisky vstupních polí
firstName=Jméno
lastName=Příjmení
```

Obrázek 8: Ukázka obsahu souboru i18n.properties [autor]

Ukázka části kódu pohledu v jazyku xml používající i18n model:

```
<content>
  <Label text="{i18n>firstName}" class="sapUiSmallMargin"/>
  <Input value="{/firstName}" valueLiveUpdate="true" width="200px" />
  <Label text="{i18n>lastName}" class="sapUiSmallMargin"/>
  <Input value="{/lastName}" valueLiveUpdate="true" width="200px" />
</content>
```

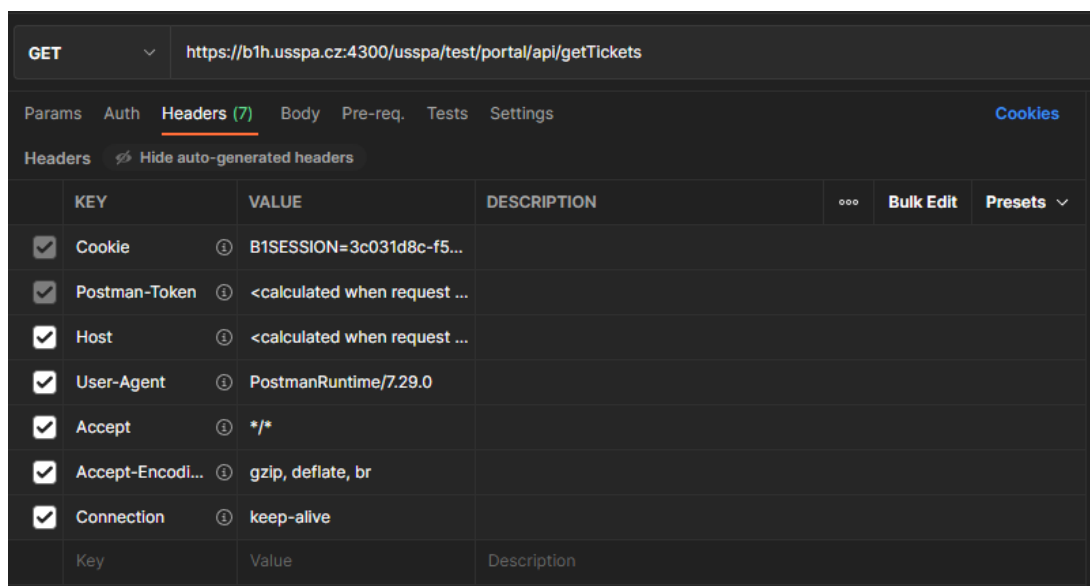
Obrázek 9: Ukázka použití modelu v jazyku xml [autor]

5 Systém pro řízení databáze SAP HANA

Požadavky podnikových aplikací se stávají čím dál náročnějšími. Tisíce uživatelů najednou mohou číst a aktualizovat záznamy. Podniky tedy potřebují využívat data okamžitě. SAP HANA v jednom databázovém systému integruje transakční i analytická data. Jedná se o in-memory databázi, tedy databázi, která je uložena v operační paměti serveru. Je sloupcově orientovaná, místo klasického řádkového uložení. Mezi přednosti systému se řadí vysokorychlostní transakce a pokročilá analýza dat. Díky tomu mohou zákazníci zpracovávat obrovské množství dat s minimální latencí. Databáze může být umístěna on-premise u zákazníka, nebo v cloudu. [10] [11]

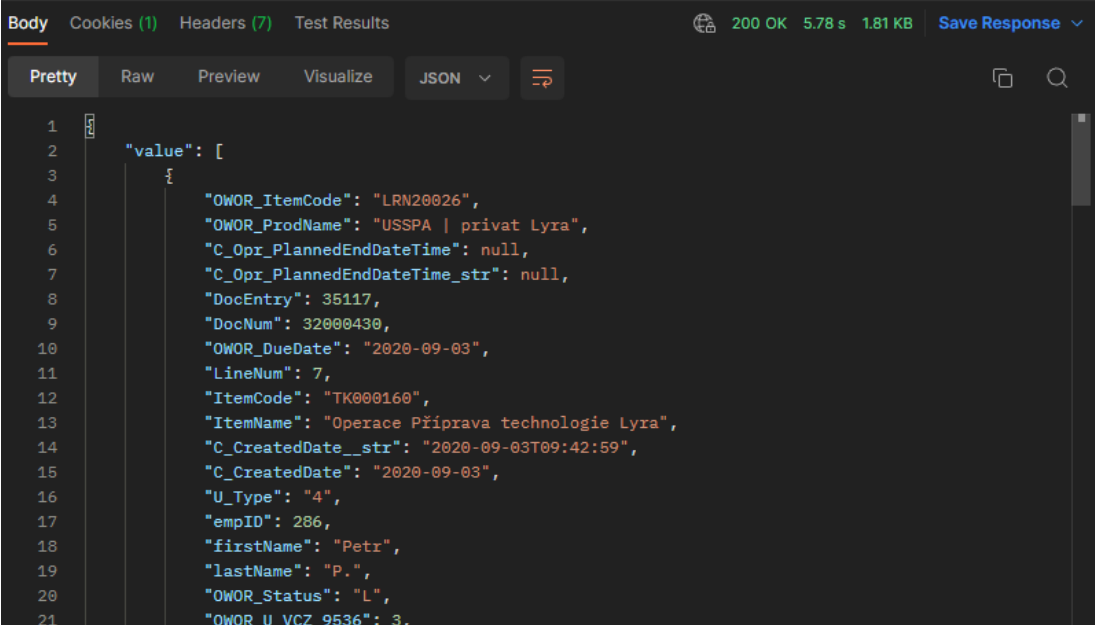
6 Vrstva služby SAP (SAP Service Layer)

Service layer je snadné provozovat v cloudovém režimu i v lokálním prostředí. Vliv na provoz nemá ani typ systému, což znamená, že nezáleží, jestli bude služba spuštěna v systému Windows nebo Linux. Tento nástroj komunikuje pomocí REST API. Načítání, přidávání a aktualizace dat probíhá pomocí metod Get, Post a Patch. Pokud chce uživatel systému SAP Business One vytvořit uživatelsky definovanou tabulku, tak jí pomocí této služby snadno vytvoří prostřednictvím rozhraní API. [12] Příklad volání služby Service layer v platformě Postman je možné vidět na Obrázku 10.



Obrázek 10: Volání metody GET [autor: aplikace Postman]

Příklad výsledku, který vrací služba Service layer v platformě Postman, je možné vidět na Obrázku 11.



```
1  [
2  "value": [
3    {
4      "OWOR_ItemCode": "LRN20026",
5      "OWOR_ProdName": "USSPA | privat Lyra",
6      "C_Opr_PlannedEndTime": null,
7      "C_Opr_PlannedEndTime_str": null,
8      "DocEntry": 35117,
9      "DocNum": 32000430,
10     "OWOR_DueDate": "2020-09-03",
11     "LineNum": 7,
12     "ItemCode": "TK000160",
13     "ItemName": "Operace Příprava technologie Lyra",
14     "C_CreatedDate_str": "2020-09-03T09:42:59",
15     "C_CreatedDate": "2020-09-03",
16     "U_Type": "4",
17     "empID": 286,
18     "firstName": "Petr",
19     "lastName": "P.",
20     "OWOR_Status": "L",
21     "OWOR_U_V CZ_9536": 3,
```

Obrázek 11: Výsledek metody GET [autor: aplikace Postman]

Nevýhody SAP Service Layer:

Jelikož je celý proces závislý na REST API, je omezen pouze na používání při připojení na internetu. [12]

7 Ukázková aplikace nad SAP (analýza a návrh)

7.1 Přehled montážních zakázek

Pro uživatele software SAP Business One byla vyvinuta aplikace Přehled montážních zakázek.

Zobrazené záznamy lze snadno filtrovat a třídít podle termínů, stavu rozpracovanosti nebo oddělení výroby (Obrázek 12).

The screenshot shows the 'Přehled montážních zakázek' application interface. At the top, there is a navigation bar with a back arrow, the title 'Přehled montážních zakázek', and a user profile icon labeled 'Demo User'. Below the navigation bar, there is a 'Kompletace' dropdown menu. The main area contains a filter bar with the following fields and controls:

- Počáteční datum:
- Termín:
- Plánované datum zahájení:
- Plánované datum dokončení:
- Oddělení:
- Třídy, skupiny:
- Montážní zakázka:
- Status:
- Typ artiklu:
- Č. produktu:
- Zákazník:
- Skrýt zabalené MZ:
- Typ montážních zakázek:
- Priorita:
- Skrýt 100% hotové:

At the bottom of the filter bar, there are buttons for 'Spustit', 'Reset', and 'Upravit filtry'. Below the filter bar, there is a summary bar showing 'Montážní zakázky (32)' and search options: 'Dohledat související: Zapnout seskupování Hledat...'. There are also icons for search, refresh, and print.

Obrázek 12: Filter bar pro tabulku [autor]

Následný výstup filtrů je možné vidět v tabulce na Obrázku 13.

The screenshot shows the 'Přehled montážních zakázek' application interface with a table of filtered data. The table has the following columns: Č. produktu, Číslo montážní zakázky, Plánované množství, Počáteční datum, Termín, Plánované datum zahájení, Plánované datum dokončení, Priorita, Status, and Rozpracovanost / Vydáný materiál. The table contains three rows of data, each with a progress bar and a filter icon.

Č. produktu	Číslo montážní zakázky	Plánované množství	Počáteční datum	Termín	Plánované datum zahájení	Plánované datum dokončení	Priorita	Status	Rozpracovanost / Vydáný materiál
SXX21102	32100513	1	3. 1. 2022	4. 1. 2022	6. 1. 2022 7:00	6. 1. 2022 7:36	Standardní	Uvolněno	R: 97% M: 91%
ARN21012	32100662	1	3. 1. 2022	4. 1. 2022	6. 1. 2022 7:00	6. 1. 2022 7:24	Standardní	Uvolněno	R: 93% M: 56%
BTP025SV501	42200001	1	3. 1. 2022	4. 1. 2022	7. 1. 2022 11:45	7. 1. 2022 12:52	Standardní	Uvolněno	R: 0% M: 20%

Below the table, there is a summary bar showing 'Montážní zakázky (32)' and search options: 'Dohledat související: Zapnout seskupování Hledat...'. There are also icons for search, refresh, and print.

Obrázek 13: Tabulka s daty [autor]

Pro vizualizaci detailu montážní zakázky stačí kliknout na řádek v seznamu. Přehledně se zobrazí nejen jednotlivé operace zakázky, ale i veškerý materiál a související polotovary (Obrázek 14).

V časové ose jsou pak přehledně zobrazeny jednotlivé operace s náhledem detailu a možností různých pohledů dle pracovišť (Obrázek 15).

Detail montážní zakázky

Číslo montážní zakázky: 32100580 Zákazník: Univerzita Hradec Králové Plánované datum zahájení: 6. 1. 2022 7:00
 Typ montážní zakázky: Zákázka oděratele: 9481 Plánované datum dokončení: 7. 1. 2022 9:27
 Priorita: Standardní Počáteční datum: 3. 1. 2022 Skutečné datum uzavření:
 Status: Uvolněno Termín: 5. 1. 2022
 Zhotovené: 0 / 1 Datum dodávky: 10. 1. 2022

Rozpracovanost: 8% Vydány materiál: 63%

Č. produktu	Číslo montážní zakázky	Plánované množství	Zakázka oděratele	Počáteční datum	Termín	Plánované datum zahájení	Plánované datum dokončení	Priorita	Status	Rozpracovanost / Vydány materiál
AU040110	52100777	1	9481	4. 1. 2022	6. 1. 2022			Standardní	Uzavřeno	100% / 100%
Popis produktu: Skořepina Persea IN - White Pearlescent Zákazník: Univerzita Hradec Králové										
WN14MCT2	22100917	1	9481	4. 1. 2022	6. 1. 2022			Standardní	Stornováno	0% / 0%
Popis produktu: Kabinět Persea IN - Thermowood masív plný s nerezovým lemem Zákazník: Univerzita Hradec Králové										
EMIL0410	42101267	1	9481	4. 1. 2022	6. 1. 2022			Standardní	Stornováno	0% / 0%
Popis produktu: Zvedák termokrytu mechanický Persea s kabinetem. Linea - Šedá, Hladká										

Obrázek 14: Detail montážní zakázky [autor]

Pohled na časovou osu

Pracoviště: [výběr] pátek, led 7, 22

Zahmout i polotovary: ANO

Popis	04:00	06:00	08:00	10:00	12:00	14:00
Plumbing CNC						
Elektro příprava 1						
Plumbing motorů 2						
Expdice 1						
Zpracování a testování vodou-...						
Očáření-do kabínety						
Vychytání						
M. Konec						
Balení						

Obrázek 15: Pohled na časovou osu [autor]

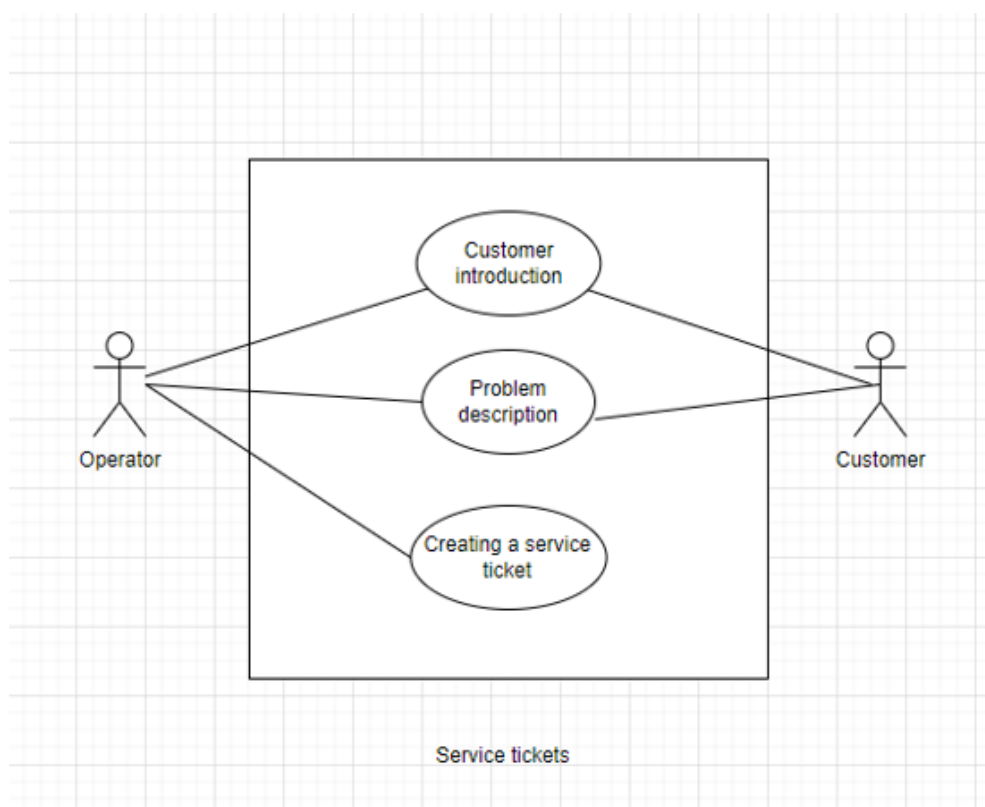
7.2 Sledování práce

Mezi další aplikaci zvolenou pro představení webových aplikací nad databází SAP je aplikace Sledování práce. Zaměstnavatelé mají v dnešní době velké možnosti týkající se kontroly činností svých zaměstnanců. Aby mohli efektivně kontrolovat činnost svých podřízených, zda řádně plní své úkoly a efektivně využívají přiřazený čas na jejich úkoly, byla vyvinuta tato aplikace. Obsluha aplikace zde může v reálném čase vyhodnocovat a schvalovat produktivitu jednotlivých zaměstnanců.

Mezi hlavní přínosy této aplikace patří zlepšení organizace výroby a s tím související mnohem lepší využití výrobní kapacity, zlepší se i organizace nákupu polotovarů a ostatního materiálu. Celkově se zvýší efektivita a ziskovost firmy.

Případy užití

Na základě funkčních požadavků byly definovány případy užití, které jsou zobrazeny na následujícím obrázku:



Obrázek 16: Use case diagram [autor]

Vzhled aplikace:

Aplikace Sledování práce zobrazuje seznam všech hlášení pracovníků při výrobě. Záznamy lze filtrovat dle zakázek, operací, pracovníků, času, oddělení a stavu (Obrázek 17). Vyfiltrovaná data se poté přehledně zobrazí v tabulce (Obrázek 18).

Kontrola práce Demo User

Výchozí filtry

Montážní zakázka: Operace: Pracovníci: Datum hlášení: Plán. datum dokončení: Status:

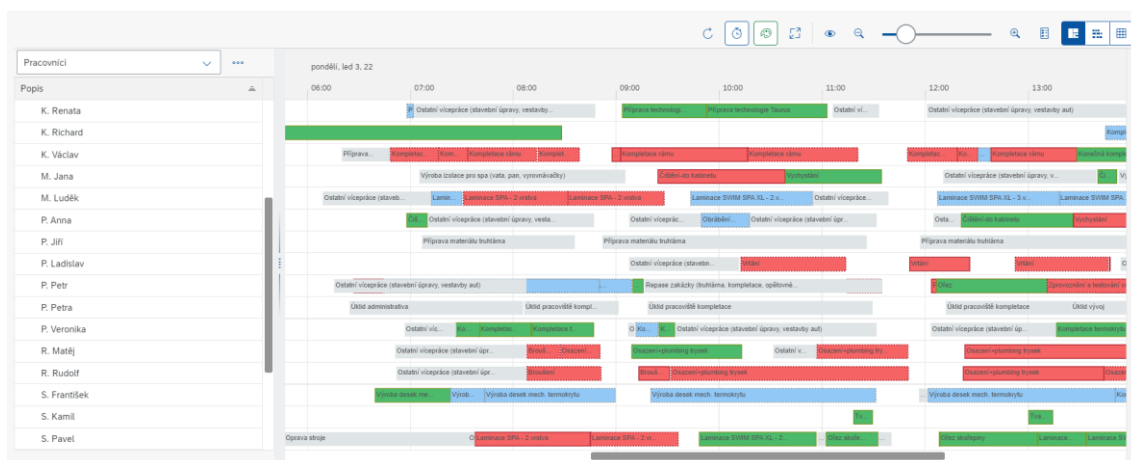
Oddělení: Třídy, skupiny: Pracoviště: Autorizováno:

Obrázek 17: Filtr pro data [autor]

Hlášení (469)										Pouze poslední hlášení: <input type="radio"/> NE <input type="radio"/> ANO		Hledat...	<input type="button" value="↺"/>	<input type="button" value="↻"/>	<input type="button" value="↑"/>	<input type="button" value="↓"/>	<input type="button" value="✓"/>	<input type="button" value="✗"/>
<input type="checkbox"/>	Č.produktu	Popis produktu	Číslo MZ	Řádka	Zákazník	Popis operace	Plánované datum dokončení	Datum hlášení	Hlášeno	Status (celkem/plán/rozdíl)	Jméno pracovníka (celkem/nahlásil)	<input type="button" value="↕"/>						
<input type="checkbox"/>	CBN21047	USSPA privat Columba	32100677	9	Univerzita Hradec Králové	Operace vrtání		3. 1. 2022 6:32:30		Probíhá (0:17 / 1:18)	Š. David (0:00)	<input type="button" value="↕"/>						
<input type="checkbox"/>	CBN21047	USSPA privat Columba	32100677	9	Univerzita Hradec Králové	Operace vrtání		3. 1. 2022 6:32:51	+0:00 hod	Dokončeno (0:17 / 1:18)	Š. David (0:00)	<input type="button" value="↕"/>						
<input type="checkbox"/>	CBN21047	USSPA privat Columba	32100677	11	Univerzita Hradec Králové	Operace osazení+plumbing trysek		3. 1. 2022 6:32:58		Probíhá (13:49 / 8:12 / 5:37)	Š. David (1:00)	<input type="button" value="↕"/>						
<input type="checkbox"/>	R-03KOM	Režijní práce KOMPLETACE	92000001	16	Univerzita Hradec Králové	Elektropříprava		3. 1. 2022 6:35:55		Probíhá (940:15 / 1:00)	G. Stáza (878:41)	<input type="button" value="↕"/>						
<input type="checkbox"/>	R-02TRUH	Režijní práce TRUHLÁRNA	92000003	2	Univerzita Hradec Králové	Příprava materiálu truhlárna		3. 1. 2022 6:37:07	+0:19 hod	Odhlášeno (8794:23 / 1:00)	Z. Bohdan (543:08)	<input type="button" value="↕"/>						
<input type="checkbox"/>	WU0610TW	Kabinet Lyra IN - Thermowood palubky	22100967	10	Univerzita Hradec Králové	Operace tučení desek		3. 1. 2022 6:37:21		Probíhá (1:13 / 1:27)	Z. Bohdan (1:13)	<input type="button" value="↕"/>						
<input type="checkbox"/>	R-02TRUH	Režijní práce TRUHLÁRNA	92000003	2	Univerzita Hradec Králové	Příprava materiálu truhlárna		3. 1. 2022 6:38:23	+0:17 hod	Odhlášeno (8794:23 / 1:00)	E. Tomáš (396:36)	<input type="button" value="↕"/>						
<input type="checkbox"/>	WU0610TW	Kabinet Lyra IN - Thermowood palubky	22100966	9	Univerzita Hradec Králové	Operace Výroba lemu		3. 1. 2022 6:38:39		Probíhá (1:28 / 1:36)	E. Tomáš (1:28)	<input type="button" value="↕"/>						

Obrázek 18: Tabulka s daty [autor]

Další obrazovka pak v časové ose zobrazuje činnosti jednotlivých pracovníků s náhledem detailu dané operace (Obrázek 19).



Obrázek 19: Ganttův diagram [autor]

7.3 Příjem požadavků na servis

Prvotním impulzem pro vznik aplikace byl požadavek na optimalizaci procesů příjmů požadavků na servis a vytvoření servisního hlášení v Business One. Původní řešení obnášelo značné množství manuální práce. Evidence požadavků na servis se udržovala mimo Business One, především v tabulkovém software Microsoft Excel. Jednalo se tedy o řešení bez jakéhokoliv propojení na aktuální data zákazníků. Po zpracování požadavků do Excelu se musely údaje ručně přepsat do určených tabulek v Business One a servisní hlášení se muselo vytvořit manuálně.

Jako nové řešení byla zvolena webová aplikace, která slouží jako přímá náhrada za dříve používaný Microsoft Excel. Mezi její výhody patří přímé napojení na SAP Business One, což obsluze umožňuje rychle zjistit všechny potřebné informace o zákazníkovi. Zákazník zavolá na infolinku firmy a obsluha na infolince začne psát do vstupního pole v aplikaci údaje o zákazníkovi podle jména zákazníka nebo podle sériového čísla produktu. Aplikace začne automaticky nabízet data, která dohledala v databázi. Obsluha aplikace si pak může vybrat z nabídky, o kterého zákazníka se jedná. Po vybrání zákazníka se do aplikace načtou detailnější data jako například adresa zákazníka, seznam kontaktních osob zakoupeného vybavení, případně další dokumenty a návody k zakoupenému produktu. Jakmile vyplníme všechny potřebné

údaje, tak se nám zpřístupní možnost vytvořit z tiketu jedním kliknutím servisní hlášení.

Hlavní stránka s tabulkou, ve které jsou zobrazeny tikety (Obrázek 20).

The screenshot shows a web application interface for service requests. At the top, there is a navigation bar with a back arrow, the text 'Přijím požadavků na servis', and a user profile 'Demo User (SBO_CZ_TEST)'. Below the navigation bar are two icons: 'Požadavky' (14) and 'Archiv'. The main content area is titled 'Seznam tiketů' and contains a table with the following data:

ID	Zpracoval	Datum příjmu	Zákazník	Kontaktní osoba	Adresa	Spa	Sériové číslo	Porucha	Volání zpět
11	Demo User	11. 10. 2021 9:29:00	UHK 721 885 446	B. Novák 648 553 876	Rokitanského 62 Brno Brno-město	ComfortCLUB Spa Arena	AR12008	-	Ne
8	Demo User	6. 10. 2021 14:46:12	UHK 721 885 446	B. Novák 648 553 876	Rokitanského 62 Pardubice Pardubice	ComfortCLUB Spa Rondo	RO10003	AA - obecné (**panýlek/displej/tlačítko nefunguje)	Ne
7	Demo User	6. 10. 2021 14:45:48	UHK 721 885 446	B. Novák 648 553 876	Rokitanského 62 Praha 9 - Horní Počernice Hlavní město Praha	Royal Spa Princezna	7440402SHK	AA - obecné (**tlačítko nefunguje)	Ne
5	Demo User Demo User	30. 9. 2021 9:20:26	UHK 721 885 446	B. Novák 648 553 876	Rokitanského 62 Brno 2 Brno-město	Universum Spa Draco	9060705	AA - obecné (**tlačítko nefunguje)	Ne

Obrázek 20. Hlavní stránka [autor]

Obsluha aplikace může vyhledávat zákazníka pomocí napovídání. Obsluha začne psát do příslušného políčka základní informace, jako například jméno zákazníka, a poté se obsluze zobrazí tabulka s detailnějšími informacemi přímo ze SAP Business One (Obrázek 21). Výběrem požadovaného řádku se poté vyplní informace do jim příslušných políček ve formuláři.

The screenshot shows a search interface for customers. It includes a search bar with the text 'Vyhledání zákazníka' and a dropdown menu for 'Typ obchodního partnera' set to 'Vše'. The search criteria are 'Jméno/Telefon' and 'Nová'. The search results are displayed in a table with the following data:

Název	Telefon	Jméno kontaktu	Telefon kontaktu	Typ
UHK	722 734 566	N. Novák	722 354 785	Zákazník
UHK	722 734 566	S. Novák	722 354 785	Zájemce
UHK	722 734 566	N. Novák	722 354 785	Zájemce
UHK	722 734 566	S. Novák	722 354 785	Zájemce
UHK	722 734 566	J. Novák	722 354 785	Zájemce
UHK	722 734 566	J. Novák	722 354 785	Zájemce
UHK	722 734 566	p. Novák	722 354 785	Zájemce
UHK	722 734 566	p. Novák	722 354 785	Zájemce
UHK	722 734 566	p. Novák	722 354 785	Zájemce
UHK	722 734 566	K. Novák	722 354 785	Zájemce

Obrázek 21: Napovídání dat [autor]

Dále může obsluha vyhledávat například podle města. Po zadání názvu města a stisknutí tlačítka vyhledat se obsluze zobrazí tabulka s informacemi. (Obrázek 22). Po vybrání požadovaného řádku se data vyplní do jim příslušných políček.

Seznam tiketů / Nový /
Nový požadavek

ZÁKLADNÍ INFO DOPLŇUJÍCÍ INFO

Ulice:

Město:

Vyber zákazníka

Název	Adresa	Spa	Sériové číslo
UHK	Rokitanského 62, Pardubice, Zákazník	Mini Spa Bellis	ZSBF S. R.
UHK	Rokitanského 62, Pardubice, Zákazník	Royal Spa Císař	8420503
UHK	Rokitanského 62, Pardubice, Zákazník	Universum Spa Columba	ZNOVOTNY JI
UHK	Rokitanského 62, Pardubice, Zákazník	Universum Spa Draco	7021965
UHK	Rokitanského 62, Pardubice 2, Zákazník	Universum Spa Lyra	14580506

Obrázek 22: Hledání podle města [autor]

Po stisknutí řádku s tiketem na hlavní stránce se uživateli zobrazí detail tiketu. Zde tiket může editovat, archivovat, nebo z něho může vytvořit servisní hlášení (Obrázek 23).

Seznam tiketů / Detail /
Tiket č.8

6. 10. 2021 14:46:12 Zákazník: Demo User Kontaktní osoba: B. Novák Vybavení: ComfortCLUB Spa Rondo Info: Volání zpět: Ne
 721 885 446 Telefon: 648 553 876 Sériové číslo: RO10003 Požadovaný termín servisu: 6. 10. 2021 9:00:00

Vytvoření servisního hlášení

Kód obchodního partnera: **Neuvedeno** Požadovaný termín servisu: **Neuvedeno**

Jméno obchodního partnera: UHK Typ termínu: Orientační

Kontaktní osoba: B. Novák Druh hlášení:

Telefon: 648 553 876 Doprava:

Artikl: **Neuvedeno**

Popis: ComfortCLUB Spa Rondo

Sériové číslo: **Neuvedeno**

ZÁKLADNÍ INFO

Zákazník
Zákazník: UHK
Telefon: 721 885 446

Kontaktní osoba
Jméno: B. Novák
Telefon: 648 553 876

Vybavení
Spa: ComfortCLUB Spa Rondo
Sériové číslo: RO10003

Obrázek 23: Vytvoření servisního hlášení [autor]

8 Ukázková aplikace nad SAP (implementace)

V této kapitole je popsána implementace webové aplikace. Součástí kapitoly je popis struktury projektu, důležité části kódu a algoritmů pro chod webové aplikace. Při vývoji aplikace byla použita architektura MVC. Implementace probíhala v nástroji Visual Studio Code od společnosti Microsoft.

Index.html je hlavní soubor, který slouží ke spuštění komponenty. Skládá se ze dvou částí head a body. Prohlížeč ke zpracování dokumentu použije část head. Můžeme v ní ovlivnit chování prohlížeče pomocí meta tagů. Načítá se zde knihovna SAPUI5, pomocí procesu, kterému se říká bootstrapping. V části body je popsáno rozložení stránky.

Component.js obsahuje definici objektu `sap.ui.core.UIComponent`, který představuje celou aplikaci jako komponentu. Vytvořené modely a funkce na tomto místě nám umožní k nim poté přistupovat ze všech pohledů i controllerů. Načítá se zde soubor `Manifest.json`.

Manifest.json obsahuje definice metadat komponenty. Je zde zahrnuta konfigurace na úrovni aplikace a informace o routingu.

Formatter.js je použit pro především složitější logiku formátování vlastností datového modelu. Programují se v něm vlastní formátovací funkce.

i18n.properties ve složce `i18n` uchovává texty pro internacionalizační model. Díky tomu, že se všechny texty UI nacházejí v jednom souboru, může být aplikace snadno přeložena do jiných jazyků.

Notifications.fragment.xml v tomto souboru se nachází fragment, který zobrazí dialog s notifikacemi.

ErrorManager.js obsahuje logiku chování v případě, že v aplikaci nastane chyba. Funkce `handleError` přijímá data z volání AJAX. Data následně formátuje a uloží do modelu. Podle nastavení pak vybere druh zobrazení chybové hlášky a zobrazí ji uživateli.

8.1 Přehled montážních zakázek

Views

HomePage.view.xml v tomto souboru je definován pohled na hlavní obrazovku, který je v této aplikaci jako výchozí. Obsahuje ovládací prvek `sap.m.App`, sloužící jako kontejner, do kterého se pomocí agregace vkládají další prvky. V něm je vnořený ovládací prvek `sap.f.DynamicPage`, ve kterém jsou vnořeny dva hlavní prvky aplikace `sap.ui.comp.filterbar`, který slouží k filtrování dat v aplikaci a `sap.m.Table`, který slouží k zobrazení dat v tabulce.

AnalyticalPage.view.xml se zobrazí po kliknutí na řádek v tabulce na hlavní stránce a zobrazí detailní informace o montážní zakázce.

SortingDialog.fragment.xml v tomto souboru je vytvořen dialog pro ovládání třídění a řazení dat v tabulce. Jako prvek je použit `sap.m.ViewSettingsDialog`.

Controllers

HomePage.controller.js obsahuje veškerou logiku, která je použita na hlavní stránce. Načítají a se zde veškerá potřebná data ze SAP Bussines One. Pomocí filterbaru se vytvoří filter na požadovaná data obsluhy aplikace, který se následně v backendu aplikace filtruje.

AnalyticalPage.controller.js zde se načítají detailnější informace o montážní zakázce, včetně jejich podřízených zakázkách, jednotlivých operací montážní zakázky i materiálu, který je u zakázky potřeba.

BasicController.js jedná se o třídu, ze které ostatní controllery dědí a obsahuje jejich společné funkce.

8.2 Sledování práce

Views

HomePage.view.xml v tomto souboru je definován pohled na hlavní obrazovku, který je v této aplikaci jako výchozí. Obsahuje ovládací prvek `sap.m.App`, sloužící jako kontejner, do kterého se pomocí agregace vkládají další prvky. V něm je vnořený ovládací prvek `sap.f.DynamicPage`, ve kterém jsou vnořeny dva hlavní prvky aplikace `sap.ui.comp.filterbar`, který slouží k filtrování dat v aplikaci.

SortingDialog.fragment.xml v tomto souboru je vytvořen dialog pro ovládání třídění a řazení dat v tabulce. Jako prvek je použit `sap.m.ViewSettingsDialog`.

GanttChart2.fragment.xml obsahuje vzhled Ganttova diagramu. Je zde použitý prvek `sap.gantt.simple.GanttChartContainer`. Ten má v sobě vnořený `sap.gantt.simple.GanttChartWithTable` s jeho vnořeným prvkem `sap.ui.table.TreeTable` pro zobrazení tabulky s daty a `sap.gantt.simple.GanttRowSettings` pro zobrazení Ganttova diagramu.

TimeSheetTable.fragment.xml obsahuje vzhled tabulky na hlavní stránce pomocí prvku `sap.m.Table`.

Controllers

HomePage.controller.js obsahuje logiku, která je použita na hlavní stránce. Načítají a se zde veškerá potřebná data ze SAP Business One. Pomocí filterbaru se vytvoří filter na požadovaná data obsluhy aplikace, který se následně v backendu aplikace filtruje.

MyGantt2.js zde se vytváří požadovaná datová struktura pro ganttův diagram.

MyGanttCfg.js jedná se o třídu, kde se vytváří konfigurace pro Ganttův diagram. Například jaké barvy budou v diagramu použity. Dále se zde vytvářejí úrovně pro `sap.ui.table.TreeTable`.

8.3 Příjem požadavků na servis

Views

TicketHomePage.view.xml v tomto souboru je definován pohled na hlavní obrazovku, který je v této aplikaci jako výchozí. Obsahuje ovládací prvek `sap.m.Page`, sloužící jako kontejner, do kterého se pomocí agregace vkládají další prvky. V něm je vnořený ovládací prvek `sap.f.DynamicPage`, ve kterém je vložený `sap.ui.comp.filterbar`, který slouží k filtrování archivu. Dále je zde vložený prvek `sap.m.IconTabBar`, který se používá pro navigaci mezi tabulkou s požadavky a archivem. Tyto dvě tabulky jsou tvořeny prvkem `sap.m.Table`.

NewTickets.view.xml slouží k vytvoření tiketu servisního hlášení. Je zde formulář s několika druhy napovídání, usnadňující vyplnění formuláře. Napovídání dat je tvořeno pomocí dat přímo ze SAP Business One.

Detail.view.xml slouží k zobrazení detailu tiketu.

Edit.view.xml slouží k editaci tiketu. Jsou zde použité stejné prvky jako v `NewTickets.view.xml`. Jako prvek je zde použit `sap.m.Dialog`.

CreateServiceReportDialog.fragment.xml tento fragment se zobrazí při stisknutí tlačítka vytvořit servisní hlášení.

Controllers

TicketHomePage.controller.js načítá data z databáze. Po načtení dat je uloží do modelu a data se následně zobrazí v tabulce.

NewTickets.controller.js slouží k vytvoření tiketu servisního hlášení. Je zde formulář s několika druhy napovídání, usnadňující vyplnění formuláře. Napovídání dat je tvořeno pomocí dat přímo ze SAP Business One.

Detail.controller.js načítá detailní informace o tiketu. Tiket je zde možné archivovat, přejít na obrazovku s editací tiketu, nebo vyvolat dialog **CreateServiceReportDialog.js**, který v případě, že jsou všechny povinné informace vyplněny, vytvoří z tiketu servisní hlášení.

Edit.controller.js slouží k editaci tiketu. Je zde stejná funkcionality, jako v `NewTickets.controller.js`

9 Dosažené výsledky

Celkem byly v této práci vytvořeny tři webové aplikace s využitím komponent z frameworku SAPUI5. Jako první práce byla vytvořena aplikace Přehled montážních zakázek. Tato aplikace umí pomocí uživatelsky nastavovatelných filtrů filtrovat montážní zakázky například podle plánovaného data zahájení nebo dokončení, podle oddělení, typu artiklu, statusu montážní zakázky a podle zákazníka. Následně přehledně zobrazí data o montážní zakázce do tabulky. Mezi zobrazená data patří například číslo montážní zakázky, její termín, status a aktuální rozpracovanost.

Jako druhá aplikace byla vytvořena aplikace Sledování práce. Tato aplikace umožňuje mistrům mít přehled nad jednotlivými pracovníky. Mohou pak schvalovat jejich výkaz práce, nebo naopak v případě nedostatečné pracovní morálky jejich výkaz práce zamítnout. Podobně jako v první aplikaci je zde možnost data filtrovat dle potřeby. Tato data mohou být zobrazena v přehledném Ganttově diagramu.

Poslední vytvořenou aplikací byla aplikace Příjem požadavků na servis. Tato aplikace slouží jako sběr tiketů servisních hlášení. Obsluha aplikace při vytváření tiketu používá připravený formulář s nutnými požadavky pro vytvoření servisního hlášení v SAP Business One. Formulář umožňuje obsluze rychle dohledávat data ze SAPu pomocí napovídání. Po vybrání hledaných dat z tabulky napovídání se další informace z databáze doplní na jejich příslušná místa ve formuláři. Poté, co je tiket vyplněný všemi povinnými údaji, je možné z něj v jeho detailu vytvořit servisní hlášení v SAP Business One.

10 Shrnutí práce

Cílem práce bylo představit tvorbu webových aplikací nad databází SAP Business One. V teoretické části byly popsány technologie využívané v praktické části této práce. K těmto technologiím patří například OData Protocol, SAP Fiori, SAPUI5, AJAX a SAP HANA. Vývoj frontendu webových aplikací nad databází SAP Business One byl následně demonstrován na tvorbě třech webových aplikací. Při vývoji těchto aplikací byl použit webový framework SAPUI5. Výsledné aplikace byly nasazeny do prostředí SAP HANA.

11 Závěr a doporučení

Všechny webové aplikace vytvořené v této bakalářské práci se v současné době používají v každodenním provozu v několika firmách. Některé aplikace nahradily dříve používané řešení v podobě tabulek v programu Microsoft Excel. Aplikace jsou dále průběžně upravovány a rozšiřovány na základě zpětné vazby a aktuálních požadavků zákazníků.

Při vývoji webových aplikací bych vývojářům doporučoval se více soustředit na hlubší analýzu požadavků zákazníka. Během vývoje aplikací v této práci bohužel nebyl dostatek časového prostoru na důkladnou analýzu. Z toho důvodu byly aplikace částečně předělávány, aby byl splněn celkový cíl.

Aplikace by mohly být dále lépe optimalizovány, aby se snadněji daly použít pro jiné zákazníky. Do budoucna by mohly být aplikace rozšířeny o automatizované testy, aby se nemusely testovat uživatelsky a tím se ušetřil další čas programátora i zákazníka.

12 Použitá literatura

- [1] MATENDE, Samwel; OGAO, Patrick. Enterprise resource planning (ERP) system implementation: a case for user participation. *Procedia Technology*, 2013, 9: 518-526. ISSN: 2212-0173
- [2] SAP Business One: What is SAP Business One? [online]. [cit. 2022-05-26]. Dostupné z: <https://www.sap.com/products/business-one.html>
- [3] Itpfed: SAP ODATA EXPLAINED [online]. [cit. 2022-05-28]. Dostupné z: <https://itpfed.com/sap-odata-explained/>
- [4] VALABOJU, Yeshwanth. Capabilities and Key Benefits of Sap NetWeaver Gateway. *International Journal of Innovative Research in Computer and Communication Engineering*, 2019, 7.1. ISSN: 2320-9801
- [5] PEKKA, Lankila. Replacing SAP Portal with SAP Fiori in Application Development. 2019. ISSN: 1799-2710
- [6] SZIRTES, Tamas; RIVLIN, Aviad. *Implementing SAP Fiori Launchpad*. 2015. ISBN: 978-1-4932-1291-4
- [7] GOEBELS, Christiane, et al. *SAPUI5: The Comprehensive Guide*. Rheinwerk Publishing, 2020. ISBN: 978-1-4932-1901-8
- [8] Kod'ousková, Barbora. ARCHITEKTURA MVC: DEFINICE, STRUKTURA, FRAMEWORKY. Rascasone [online]. [cit. 2022-05-28]. Dostupné z: <https://www.rascasone.com/cs/blog/architektura-mvc-struktura-frameworky>
- [9] W3schools: AJAX Introduction [online]. [cit. 2022-06-10]. Dostupné z: https://www.w3schools.com/js/js_ajax_intro.asp
- [10] ANDREI, Mihnea, et al. SAP HANA adoption of non-volatile memory. *Proceedings of the VLDB Endowment*, 2017, 10.12: 1754-1765. ISSN: 2150-8097

[11] Sap: What is SAP HANA? [online]. [cit. 2022-05-26]. Dostupné z:

<https://www.sap.com/products/hana/what-is-sap-hana.html#overview>

[12] Routh, Subhasish. Advantages and Disadvantages: Service Layer for SAP Business One Integration. Appseconnect [online]. [cit. 2022-06-14]. Dostupné z:

<https://www.appseconnect.com/advantages-and-disadvantages-of-using-service-layer-for-sap-business-one-integration/>

13 Přílohy

Soubor README.txt, který obsahuje odkaz na zdrojové kódy aplikace, dále obsahuje odkaz, kde je možné webovou aplikaci vyzkoušet a přístupové údaje do aplikace.

Zadání bakalářské práce

Autor:	Michal Ludvíček
Studium:	I1900675
Studijní program:	B0688A140001 Informační management
Studijní obor:	Informační management
Název bakalářské práce:	Webové aplikace nad databází SAP
Název bakalářské práce AJ:	Web applications over SAP database

Cíl, metody, literatura, předpoklady:

Cílem bakalářské práce je představit postup tvorby webových aplikací nad databází SAP a demonstrovat jej nad ukázkovou aplikací

Struktura práce:

1. Úvod
2. Cíl práce, metodika
3. SAP: Systém pro plánování podnikových zdrojů
4. Framework pro uživatelské rozhraní SAP UI5
5. Systém pro řízení databáze SAP HANA
6. Vrstva služby SAP (SAP Service Layer)
7. Ukázková aplikace nad SAP (analýza a návrh)
8. Ukázková aplikace nad SAP (implementace)
9. Dosažené výsledky
10. Shrnutí práce
11. Závěr a doporučení
12. Použitá literatura

LEWIS, Carl, Andres CASTRILLON, Derin HILDEBRANDT, Ryan HOWE, Keith TAYLOR a Bertrand TOUGAS. SAP Business One: Business User Guide. 2017. ISBN 978-1-4932-1500-3.

VAN KEMPEN, Denys. SAP HANA 2.0: An Introduction. 2019. ISBN 978-1-4932-1839-4.
ALBORGHETTI, Francesco, Jonas KOHLBRENNER, Abani PATTANAYAK, Dominik SCHRANK a Primo SBOARINA. SAP HANA XSA: Native Development for SAP HANA. 2018. ISBN 978-1-4932-1602-4.
MODDERMAN, Paul, Christiane GOEBELS, Denise NEPRAUNIG a Thilo SEIDEL. SAPUI5: The Comprehensive Guide. 2020. ISBN 978-1-4932-1901-8.

BRANDEIS, Jörg. SQLScript for SAP HANA. 2021. ISBN 978-1-4932-2140-0.

<https://sapui5.hana.ondemand.com/#/topi>

Zadávající pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Daniela Ponce, Ph.D.
Oponent:	doc. Ing. Pavel Čech, Ph.D.
Datum zadání závěrečné práce:	31.5.2022