UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

FAKULTA PRÍRODNÝCH VIED

Komunikácia mikro-frameworkov s mobilnými aplikáciami

DIPLOMOVÁ PRÁCA

Študijný program: 9.2.9 Aplikovaná informatika

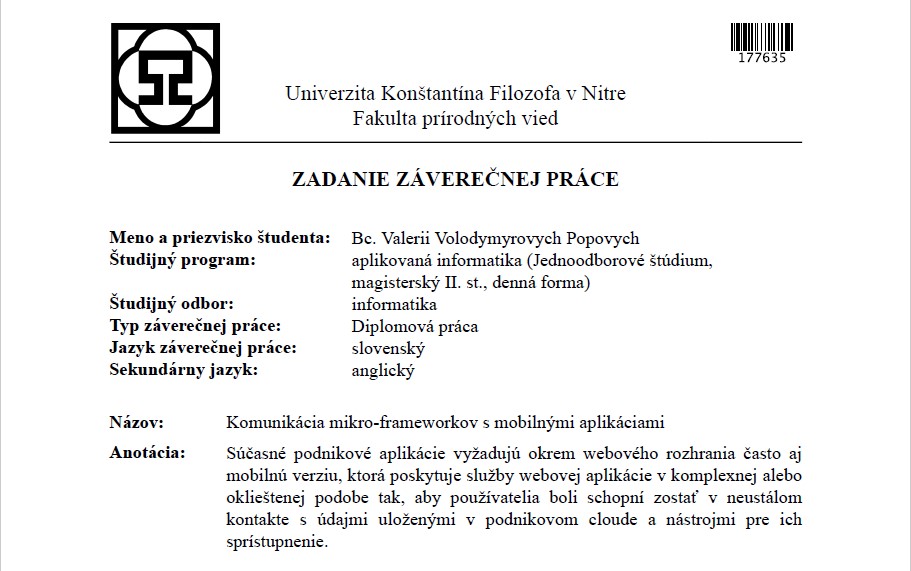
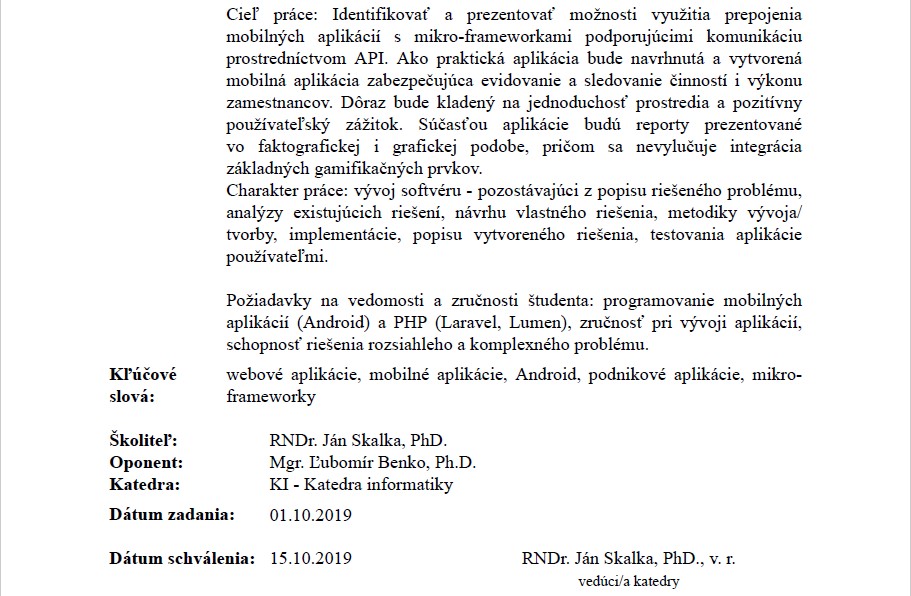
Študijný odbor: Aplikovaná informatika

Školiace pracovisko: Katedra informatiky

Školiteľ: RNDr. Ján Skalka, PhD.

Oponent: Mgr. Ľubomír Benko, PhD.

Nitra 2021 Bc. Valerii Popovych

**ABSTRAKT**

POPOVYCH, Valerii: Komunikácia mikro-frameworkov s mobilnými aplikáciami. [Diplomová práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied. Školiteľ: RNDr. Ján Skalka, PhD. Stupeň odbornej kvalifikácie: Magister odboru Aplikovaná informatika. Nitra: FPV, 2021. s.

V tejto praci sa venujeme trendom a typom podnikovych aplikacii a pouzitie mikro frameworkov v takych typoch aplikacii. Snazime sa pochopit ake rchitektury sa pouzivaju pri vyvoji takych aplikacii. Hlavnym cielom nasej prace je navrh a vyvoj mobilnej podnikovej aplikacii pre operacny system Android. Vysledna aplikacia bude sluzit na evidenciu pracovnych vykazov zamestnancov UKF, pracovneho casu, projetov a publikacii. Malo by to byt pohodlne a jednoduche riesenie pre kazdeho zamestnanca. Taktiez taka aplikacia dovoli veducim samostatnych pracovisk mat jednoduchy prehlad vykonanej cinnosti na oddelenii/katedre/fakulte.

**Kľúčové slová:**  Podnikové aplikácie. Mikro-framework. Mobilne aplikácie. Android.

**ABSTRACT**

POPOVYCH, Valerii: Micro-frameworks Communication with Mobile Applications. [Master thesis]. Constantine the Philosopher University in Nitra. Faculty of Natural Sciences. Supervisor: RNDr. Ján Skalka, PhD. Degree of Professional Qualification: Master at Department of Applied Informatics. Nitra: FNS, 2021. p.

**Key words:** Enterprise applications. Micro-framework. Mobile applications. Android.

Obsah

[Úvod 7](#_Toc69326295)

[1 Typy a trendy podnikových aplikácií v súčasnosti 8](#_Toc69326296)

[1.1 Typy a trendy podnikovych aplikacií 8](#_Toc69326297)

[1.1.1 ERP system 8](#_Toc69326298)

[1.1.2 Manazmentove - ekonomicke aplikacie 10](#_Toc69326299)

[1.1.3 Vzdelavacie aplikacie 13](#_Toc69326300)

[1.1.4 Komunikačne aplikacie 17](#_Toc69326301)

[1.1.4.1 Zoom 18](#_Toc69326302)

[1.1.4.2 Google Meet 18](#_Toc69326303)

[1.1.4.3 Microsoft Teams 20](#_Toc69326304)

[1.2 Mobilne podnikove aplikacie 21](#_Toc69326305)

[1.3 Hlavne vývojové trendy vytvorenia dnesnych bisnes aplikacií 22](#_Toc69326306)

[2 Architektúra aplikácií 27](#_Toc69326307)

[2.1 Popis architektur 27](#_Toc69326308)

[Model a View 27](#_Toc69326309)

[2.1.1 Architektura MVC 28](#_Toc69326310)

[2.1.2 Architektura MVP 29](#_Toc69326311)

[2.1.3 Architektura MVVM 30](#_Toc69326312)

[2.2 Optimalizácia a výkon 30](#_Toc69326313)

[3 Existujuce frameworky pre mikroservis 32](#_Toc69326314)

[3.1 Definícia pojmu 32](#_Toc69326315)

[3.1.1 Vlastnosti mikro-frameworkov 34](#_Toc69326316)

[3.2 Lumen 34](#_Toc69326317)

[3.3 HLEB 36](#_Toc69326318)

[3.4 Slim 37](#_Toc69326319)

[4 Ciele záverečnej práce 39](#_Toc69326320)

[5 Návrh aplikácie 40](#_Toc69326321)

[5.1 UML navrh 40](#_Toc69326322)

[5.2 Navrh prihlasovacej obrazovky 40](#_Toc69326323)

[5.2.1 Proces autorizacie uzivatela 41](#_Toc69326324)

[5.2.2 Proces registacie uzivatela 41](#_Toc69326325)

[5.3 Navrh domovskej obrazovky 42](#_Toc69326326)

[5.4 Navrh obrazoviek so zoznamom 43](#_Toc69326327)

[5.4.1 Zoznam fakult a jednotlivych pracovisk 43](#_Toc69326328)

[5.4.2 Zoznam katedier a inych pracovisk 43](#_Toc69326329)

[5.4.3 Zoznam zamestnancov 43](#_Toc69326330)

[5.4.4 Zoznam publikacií 43](#_Toc69326331)

[5.4.5 Zoznam projektov 43](#_Toc69326332)

[5.4.6 Zoznam vykazov 43](#_Toc69326333)

[5.5 Endpointy 45](#_Toc69326334)

[5.5.1 Backend rozhranie 46](#_Toc69326335)

[Záver 48](#_Toc69326336)

[Zoznam BIBLIOGRAFICKÝCH ODKAZOV 49](#_Toc69326337)

[Zoznam príloh 51](#_Toc69326338)

# Úvod

V súčasnej dobe skoro každý ma niekoľko elektronických zariadení ktoré majú prístup na Internet, vo väčšine sú to mobilne zariadenia typu tablet, smartfón. Človek vždy ma k dispozícii aspoň jedno z týchto zariadení a používa ho na rôzne účely. Chceli by sme dokázať že typ takéhoto zariadenia nám pomôže aj pri práci a presne na evidenciu nejakých osobných informácií a pracovnej činnosti.

Posledné roky vývojári mobilných aplikácií snažia sa používať pre svoje riešenia backendove frameworky ktoré sa potom jednoducho da napojiť na vlastne frontendové riešenie. Našim cieľom bude popísať problematiku a už existujúce riešenia podnikových aplikácií. Pokúsime sa pochopiť aké hlavne štruktúry a systémy sa používajú pri vývoji takých aplikácii. Ešte jednou neoddeliteľnou súčasťou nasej prace bude porovnať existujúce mikro frameworky, pochopiť princíp ich fungovania, a potom sa pokúsiť implementovať pomocou jedného z nich naše backend riešenie.

Následne nadobudnuté nami informácie využijeme pri implementácii nášho frontendového riešenia, a to aplikácie pre platformu Android. Aplikácia bude slúžiť ako portál pre UKF, na evidenciu činnosti zamestnancov a to pracovného času, osobných a spoločných projektov a publikácii.

# 1 Typy a trendy podnikových aplikácií v súčasnosti

## Typy a trendy podnikových aplikácií

Podnikové aplikácie sa stávajú bežným podporným nástrojom pracovníkov. Užívateľ od nich čoraz viac očakáva rýchlu dostupnosť, jednoduché ovládanie, obojsmernú komunikáciu a univerzálnosť. Také špecializované riešenia ako sú: CRM (riadenie zákazníckych vzťahov), SCM (riadenie dodávateľských vzťahov), ERP (plánovanie podnikových zdrojov) a tď, sa stávajú neoddeliteľnou súčasťou podnikových aplikácií. ERP je tzv. jadro ktoré je zamerane na riadenie vnútorných podnikových procesov. CRM – systém ktorý obsluhuje procesy smerujúce k zákazníkom. SCM – systém zodpovedný za riadenie dodávateľského reťazca. Štandardne informačne systémy už v dnešnej dobe nepostačujú na pokrytie všetkých procesov rýchle sa rozvíjajúcej spoločnosti.

Každý z nás používa smartphone, tento pohodlný a kompaktný asistent nahradil desiatky rôznych druhov technológií a pomáha nám takmer vo všetkých životných procesoch. Práca a štúdium, cestovanie, zábava, voľný čas. Tak či onak, toto bolo možné vďaka dynamickému rozvoju trhu s mobilnými aplikáciami. Práve programy pre smartfóny, nám umožňujú vykonávať určité úlohy bez veľkého úsilia. Budeme hovoriť o jednom z najsľubnejších segmentov trhu s mobilným softvérom, a to o podnikových aplikáciách, ktoré používame takmer každý deň. V širšom zmysle ide o také aplikácie, ktoré sú mobilnými programami, ktoré pomáhajú organizovať náš čas, monitorovať a analyzovať rôzne pracovné procesy a upravovať naše činnosti v súlade so stanovenými normami a bariérami. Pod predložený koncept teda spadá obrovské množstvo programov.

Môžeme konštatovať, že mobilné aplikácie pre podnikateľov sú dnes jedným z najsľubnejších segmentov vývoja mobilného softvéru.

### ERP systém

ERP je podnikový informačný systém, ktorý je určený na riadenie firmy, plánovanie a organizovanie chodu podniku. ERP dôkaze vyriešiť skoro všetko začínajúc nákupom alebo predajom potrebných produktov až po riadenie vzťahmi so zákazníkmi, zamestnancami a financiami.



V súčasnosti najčastejšie ERP systémy majú podobu cloudových služieb. V takomto prípade firma musí platiť prístup pre každého svojho zamestnanca, ale výhodou v tom to prípade je to že firma nemusí riešiť potrebný hardvér, aktualizácie, bezpečnosť aj zálohovanie. Ďalšou výhodou cloudového informačného systému je prístup možný z akéhokoľvek miesta. ERP softvéry sú veľmi nápomocne zamestnancom. Väčšina zamestnancov nemusia používať jednotlivé programy ale vedia použiť moduly ERP programu ktorý zahŕňa väčšinu funkcii napríklad na evidenciu nákupov, na posielanie výplat, na evidenciu majetku, ktoré môžu naraz využívať niekoľko zamestnancov na rôznych pracovných pozíciách. Pri väčšine systémov ktoré sú na trhu samozrejmosťou je aj použitie ich na mobilných zariadeniach. Pre majiteľov podnikov je to veľkou výhodou nakoľko väčšina zamestnancov majú moderny smartfón, v takom prípade je možnosť sledovania plnenia úloh zamestnancov a ďalších ich činnostnej.

Výhody ERP systémov je:

* Optimalizácia nákladov.
* Automatizácia činnosti zamestnancov.
* Zvýšenie tržieb.
* Prehľadnosť v úlohách, dokumentoch a tď.
* Predpovede a analýzy.

Nevýhodami ERP systémov sú:

* Náklady na zavedenie ERP systému.
* Komplikovanosť systému.
* Doba zavedenia ERP systému do firmy.

Na zaver môžeme povedať že daný systém nestojí na mieste a neustále sa rozvíja. Príkladom je vývoj systému ERP III, ktorý zahŕňa medzi svoje moduly aj dnes najčastejšie využívane sociálne siete. V dnešnej dobe spôsoby komunikácie medzi ľuďmi a hlavne so zákazníkmi neustále rastu. Hlavne spôsoby komunikácie už dávno nie sú len prostredníctvom mailu a telefónu. Dnes najlepšie riešenie sú sociálne siete a hlavne ich Messengeri. ERP systémy taktiež boli ocenene mnohými firmami počas pandémie spojenej s koronavírusom, pretože viacero zamestnancov museli zostať doma a vykonávať pracovnú činnosť formou homeoffisu. Aj v tomto prípade bola komunikácia medzi zamestnancami a zákazníkmi jedným z hlavných bodov.

Teraz chceme rozdeliť podnikové aplikácie do niekoľkých kategórii, porovnať ich, popísať princíp fungovania a implementácie, a skúsiť ich porovnať.

### Manažmentove - ekonomické aplikácie

Prudký rozvoj technológií a robotická automatizácia rutinných procesov výrazne uľahčili podnikanie bez ohľadu na rozsah spoločnosti. Softvér na vykonávanie obchodných úloh sa dnes stal nevyhnutným nástrojom úspešného podnikania. Mnoho moderných spoločností teda využíva program SAP ERP vytvorený na automatizáciu účtovníctva, plánovania, monitorovania a riadenia zdrojov v podnikoch bez ohľadu na ich veľkosť a vlastníctvo. Zavedenie takýchto systémov nám umožňuje optimalizovať obchodné procesy a zbaviť zamestnancov rutinnej práce. Vedúci pracovníci v podnikaní môžu pomocou softvéru monitorovať aktivity a vykonávať úpravy, aby mohli pracovať efektívnejšie a dosahovať ciele. (NASTROYVSE, ????)

SAP - je to automatizovaný systém, ktorý ponúka súbor riešení pre budovanie spoločného informačného priestoru na základe podniku a efektívneho plánovania zdrojov a pracovných procesov. Jeho nástroje je možné používať jednotlivo aj v kombinácii. Hlavná vec je, aby prostredie bolo jednotné - to zabezpečí maximálnu efektivitu vedenia a aktualizácie údajov medzi rôznymi oddeleniami alebo inými funkčnými jednotkami spoločnosti. Aby sa podnikanie rozvíjalo najlepším možným spôsobom, je potrebné automatizovať účtovníctvo v podnikoch bez ohľadu na ich veľkosť a formu vlastníctva. Je to tiež nevyhnutné pre vedúcich spoločností, ktorí môžu rýchlo vykonať nápravné zmeny na dosiahnutie optimálneho výkonu. Účtovný softvér SAP je automatizovaný systém, ktorý umožňuje plánovať zdroje veľkých podnikov, umožní vypočítať všetko do najmenších detailov a zároveň vytvorí jediný informačný priestor.

Vďaka modulárnemu princípu fungovania ERP systému bolo možné využívať nielen jednotlivé komponenty systému SAP, ale aj ich kombinácie. ERP je miesto, kde spoločnosti ukladajú základné prevádzkové údaje. Hovoríme o prognózach predaja, objednávkach, skladových zásobách a procesoch, ktoré sa spúšťajú na základe týchto údajov (napríklad platby dodávateľom pri objednávaní). V určitom zmysle je ERP „mozgom“ spoločnosti - ukladá všetky dôležité údaje a všetky akcie, ktoré sú týmito údajmi iniciované, do pracovných tokov. (ABDUL NIMERI, 2020)

Maximálny efekt z používania systému je možné dosiahnuť, iba ak podnik vykonáva všetky operácie v jednom informačnom prostredí. Systém SAP ERP nám umožňuje okamžite aktualizovať a odosielať údaje, ktoré okamžite smerujú do všetkých potrebných oddelení podniku.

Systém používa model s tromi tzv. vrstvami:

* klient;
* aplikačný server;
* systém, ktorý spravuje rôzne databázy.

Koncový používateľ predpokladá, že účtovný systém spojil hlavné funkcie v dvoch oblastiach:

* výkazníctvo a účtovníctvo, ktoré obsahuje všetko potrebné. Program tiež berie do úvahy náklady na výrobu vo všetkých vznikajúcich miestach, umožňuje spravovať objednávky a peniaze a brať do úvahy všetky ostatné výsledky;
* logistika umožňuje kombinovať nielen plánovanie, ale aj správu a predaj, ktorý zahŕňa fakturáciu, ďalší predaj a prepravu. Súčasťou logistiky je aj logistika umožňujúca nákup, kontrolu účtov a správu zásob.

Ako funguje účtovný systém SAP, čo to je

Toto je celý komplex riešení s nasledujúcimi funkciami:

* automatizácia práce účtovníka;
* zjednodušenie obchodných a skladových operácií;
* uľahčenie účtovníctva personálu, financií, podielov a iných aktív;
* modernizácia logistiky;
* vypracovanie najviditeľnejších rozvrhov platov.

Softvér vo všeobecnosti urýchľuje spracovanie údajov a umožňuje pomocou neho optimalizovať vývoj spoločnosti, ale bude efektívny iba s logickou štruktúrou organizácie a s dobrými vzťahmi medzi jej rôznymi oddeleniami.

Funkčné oblasti programu SAP pozostávajú z nasledujúcich modulov:

* Modul PP - plánuje a riadi také typy výroby, ako je diskrétny a nepretržitý cyklus;
* Modul MM - spravuje materiálové toky, obsahuje adresár služieb a materiálov, systém obstarávania, ktorý obsahuje prijímanie objednávok až po príjem zásob;
* Moduly AM a AA - zabezpečujú príjem, účtovníctvo, odpisy a prevod dlhodobého majetku v štruktúre podniku;
* Modul FI - je zodpovedný za financie;
* Modul PM - poskytuje technickú údržbu, ako aj ďalšie opravy zariadení;
* Modul CO - poskytuje kontrolu nad implementáciou manažérskeho účtovníctva;
* Modul SD - riadi procesy predaja systému, umožňuje fakturáciu, výdaj a prepravu;
* Modul HR - umožňuje správu a evidenciu personálu, prijímať zamestnancov a počítať mzdy.



Aké sú hlavne výhody a nevýhody softvéru SAP

V zozname výhod:

* Automaticky zvyšovať efektivitu podriadených.
* Pokrýva široké spektrum úloh - pokrýva všetky potreby veľkého podniku.
* Kombinuje sa pohodlne a prirodzene s inými kancelárskymi aplikáciami.
* Vysoké zameranie na pokročilé štandardy a výsledky, a tým aj na relevantnosť.

Medzi nevýhody patri:

* Pomerne vysoké náklady - nákup tohto softvéru sa z dlhodobého hľadiska oplatí.
* Adaptabilita je možná iba pri individuálnom ladení - ak sa niektoré funkcie nezhodujú s činnosťami spoločnosti, bude potrebne minúť ďalšie peniaze na ich preorientovanie.
* Potreba obsluhy programu špecialistami a školenia zamestnancov podniku na prácu s ním.

Ako vidíme, nevýhody sú relatívne a veľmi závisia od toho, kto použije rovnaký ERP. (CLEVERENCE, 2020)

Ako druhy príklad podobnej ERP aplikácií chceme uviesť podnikový informačný systém Money S4. Dnes svet je plný informácií, podnikanie sa stáva náročnejšie a stále je ťažšie získať konkurenčnú výhodu. Správne informácie v správnom čase sú dnes kľúčom k úspechu. Prosperita firmy záleží na rýchlosti spracovania dát a na schopnosti získať z nich rozhodujúce informácie, ktoré umožnia nielen včas reagovať na zmeny v okolí, ale do určitej miery i tieto zmeny predvídať. Logicky sa preto zvyšujú nároky na podnikové informačné systémy (ERP). Money S4 komplexným informačným systémom, pokrývajúcim požiadavky stredných a menších spoločností všetkých zameraní.

Money S4 je ideálny informačný systém určený pre spoločnosti, ktoré očakávajú vysokú mieru prispôsobivosti. Systém má veľa možností pre nastavenie systému. A vďaka tomu koncova konfigurácia zodpovedá potrebám zákazníka. Dizajn systému podporuje jednoduchú ovládateľnosť. Inštalácia aplikácie aj SQL servera je automatizovaná, vyžaduje však zručnosti administrácie systému. Dáta sú spracovávané bezpečne vďaka technológii MS SQL. Money S4 je veľmi vhodný aj pre spoločnosti, ktoré očakávajú vlastnosti veľkých ERP systémov, ale v súčasnej dobe si nemôžu dovoliť takú vysokú investíciu.

### Vzdelávacie aplikácie

Ďalšou zaujímavou pre nás kategóriou sú vzdelávacie aplikácie, ktoré každý deň nám pomáhajú vo vyučovacích procesoch, sú dobre ako pre učiteľov tak aj pre študujúcich. V poslednej dobe sa objavujú také nové koncepty ako vzdelávacia platforma alebo virtuálne výučbové prostredie, ktoré čoraz viac využívajú špecialisti v oblasti informatizácie vzdelávania. Riadené vzdelávacie prostredie je zamerané viac na proces automatizácie pracovného toku a riadenia vzdelávacej inštitúcie. Tento softvérový balík obsahuje aj elektronické vzdelávacie nástroje, informačné vzdelávacie zdroje, nástroje na zasielanie úloh, testovanie, spracovanie výsledkov vzdelávacích aktivít a riadenie vzdelávacieho procesu. Moderný koncept vzdelávacej platformy integruje širokú škálu rôznych možností. Vzdelávacia platforma je integrovaným konceptom, na automatizáciu riadenia vzdelávacej inštitúcie, virtuálne a riadené vzdelávacie prostredia. Hlavnými úlohami vzdelávacej platformy sú: organizácia vzdelávacieho procesu na základe nástrojov IKT, implementácia interaktívnej informačnej interakcie medzi študentom, učiteľom a systémom na miestnej a globálnej úrovni, automatizácia toku dokumentov a vzdelávacie aktivity vzdelávacej inštitúcie. (LEKTSII, 2015)

Systém ktorý nás najviac zaujal je Moodle. Chceme teda porovnať a posiať jeho hlavne funkcie. Moodle (modulárne objektovo orientované dynamické výučbové prostredie) je to bezplatný systém riadenia výučby zameraný predovšetkým na organizovanie interakcie medzi učiteľom a vzdelávanými, aj keď je vhodný aj na organizovanie tradičných dištančných kurzov a na podporu denného štúdia. Výukové prostredie je možné použiť na akomkoľvek počítači alebo modernom mobilnom zariadení s prístupom k sieti. Učiteľ môže pomocou Moodle vytvárať kurzy a napĺňať ich obsahom vo forme textov, pomocných súborov, prezentácií, dotazníkov atď. Ak chceme používať Moodle, stačí mať akýkoľvek webový prehliadač, vďaka ktorému je používanie tohto učebného prostredia vhodné pre učiteľa aj pre študentov. Na základe výsledkov študentov pri plnení úloh môže učiteľ hodnotiť známkami a komentovať. Moodle je teda tiež centrom pre tvorbu vzdelávacích materiálov a poskytovanie interaktívnej interakcie medzi účastníkmi vzdelávacieho procesu.

Moodle je webovo orientované prostredie. Pre svoju prácu potrebuje webový server s podporou PHP a databázový server. Moodle patrí do triedy LMS (Learning Management System) - systémy riadenia výučby. Takýto softvér častejšie sa nazýva systém dištančného vzdelávania (LMS), pretože práve pomocou týchto systémov sa dištančné vzdelávanie organizuje na mnohých univerzitách. Moodle je bezplatný softvér s licenciou podľa GPL, ktorá umožňuje používať systém zadarmo a bezbolestne ho upravovať v súlade s potrebami vzdelávacej inštitúcie a integrácie s inými produktmi.

Moodle umožňuje navrhovať, vytvárať a ďalej spravovať zdroje informačného a vzdelávacieho prostredia. Rozhranie systému bolo spočiatku zamerané na prácu učiteľov, ktorí nemajú hlboké znalosti v oblasti programovania a správy databáz, webových stránok a pod. Systém má užívateľsky prívetivé intuitívne rozhranie. Učiteľ môže samostatne, iba pomocou systému vytvoriť elektronický kurz a riadiť jeho. (VALENTINA MANYAKHINA, ARTYOM ZOLOCHEVSKY, 2016)

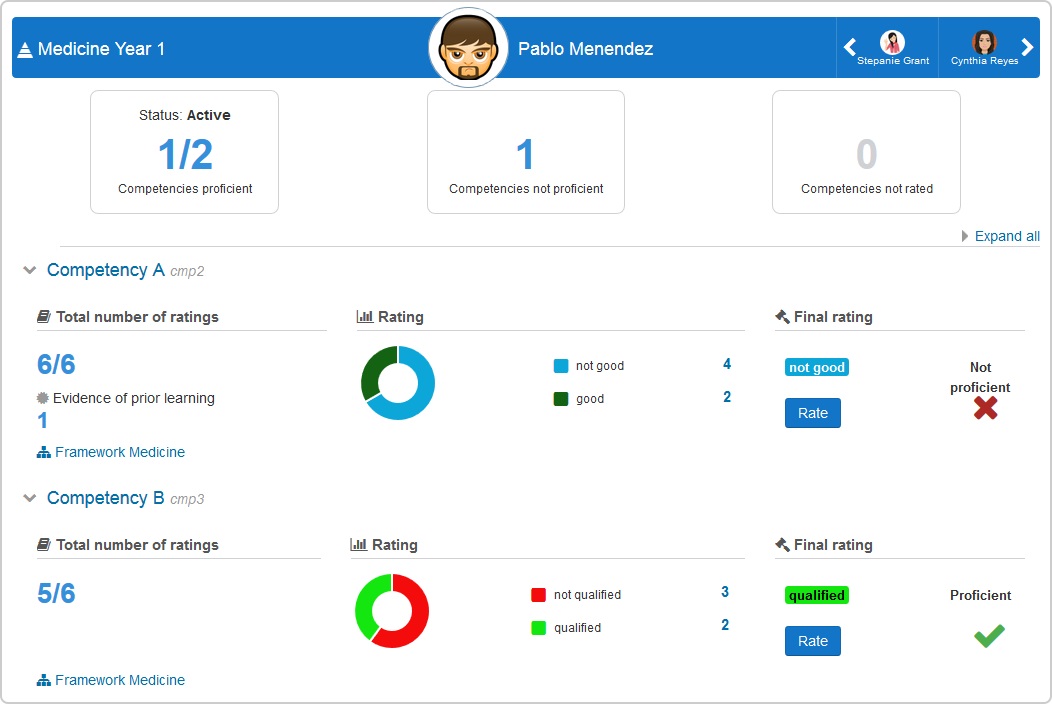
Správa vzdelávacieho procesu je dobre premyslená. Učiteľ s administrátorskými právami môže registrovať ďalších učiteľov a vzdelávaných, prideľovať im príslušné roly (tvorca kurzu, učiteľ s alebo bez práv na úpravy, vzdelávaný, hosť), distribuovať práva, kombinovať vzdelávaných do virtuálnych skupín, dostávať súhrnné informácie o práci každého vzdelávaného. Pomocou zabudovaného kalendára určiť počiatočný a konečný dátum kurzu, dodanie určitých úloh, termín testovania. Pomocou nástroja Vysvetliť a Fórum je možne zverejnenie informácie o kurze a nove správy.

Systém Moodle Learning Management System, zameraný na dištančne vzdelávanie, má širokú škálu komunikačných nástrojov. Nejde len o e-mail a výmenu príloh s učiteľom, ale aj o fórum (všeobecné správy na hlavnej stránke programu, ako aj rôzne súkromné ​​fóra), chat, súkromné ​​správy, blogovanie.

Ešte jednou neoddeliteľnou súčasťou ktorá nás najviac zaujíma je, systém výkazníctva a analýzy. Moodle má zabudovaný analytický systém, ktorý umožňuje generovať správy o činnosti na platforme. Napríklad zobrazenia kurzov, komentáre, vstupy a výstupy. Stiahnuť výkazy nie je možne, oni sa zobrazujú iba v Moodle.

Základné možnosti sú rozšírené o 38 doplnkov na generovanie štatistík. Tu sú niektoré z nich:

* My Feedback - generuje podrobné správy o pokroku v tabuľkovej forme.
* Overview Statistics zobrazuje rôzne typy prehľadov vo forme grafov.
* Device Analytics poskytuje informácie o zariadeniach, z ktorých sa používatelia prihlasujú do Moodle.
* Monitoring Learning Plans poskytuje šablóny učebných plánov a umožňuje zobrazovať ich štatistické údaje



* Events Graphic Reports vytvárajú hodnotenia vo forme grafov udalostí v Moodle

Pomocou doplnkov teda vieme nastaviť špecializovaný analytický systém. (RUSLAN YAKUPOV, 2018)

Výhody Moodle

* Bohatá funkčnosť a jednoduché použitie na ktorejkoľvek strane vzdelávacieho procesu.
* Každý poslucháč môže zvýšiť alebo znížiť mieru prezentácie materiálu, zvoliť si vhodný čas na štúdium a meniť obsah tém.
* Všetky úspešne absolvované materiály, ako aj testy s komentármi učiteľa, sa ukladajú do systému. Neskôr sa k nim môžete podľa potreby vrátiť.
* Znalosti sa lepšie získavajú vďaka inovatívnym technikám ukotvenia.
* Funkčnosť platformy možno ľahko rozšíriť tak, aby vyhovovala potrebám konkrétnej skupiny študentov alebo jedného samostatného študenta. (JEDNÉ CENTRUM VYSOKOŠKOLSKÉHO VZDELÁVANIA, ????)

Záver

Moodle je otvorený vzdelávací systém. Platforma Moodle je teda systém na vytváranie vzdelávacieho informačného prostredia vzdelávacej inštitúcie, zameraný predovšetkým na zabezpečenie interaktívnej interakcie medzi účastníkmi vzdelávacieho procesu, slúži na organizáciu externého a diaľkového vzdelávania a na podporu denného vzdelávania. Je atraktívna, pretože je zadarmo, integruje sa s desiatkami služieb a má veľký rozvojový potenciál. Počiatočná verzia má však slabú funkčnosť a je potrebné vyhľadať, inštalovať a konfigurovať ďalšie moduly a nie všetky sú zadarmo. A bez týchto funkcií možno platformu len ťažko nazvať plnohodnotným vzdelávacím systémom. Preto je Moodle zadarmo iba podmienečne: jeho implementácia si vyžaduje odborníkov a čas. Ako ukazuje prax, veľké spoločnosti uprednostňujú hotové riešenia a rýchlu implementáciu. Moodle to bohužiaľ nedokáže. (RUSLAN YAKUPOV, 2018)

### Komunikačne aplikácie

Komunikácia je vlastne komunikácia, prenos informácií od osoby inej osobe, alebo ešte môžeme povedať že je to druh interakcie medzi ľuďmi vrátane výmeny informácií. Hlavnou funkciou komunikácie je dosiahnuť sociálne spoločenstvo pri zachovaní individuality každého z jeho prvkov. Akcie vedome zamerané na ich sémantické vnímanie sa nazývajú komunikatívne. Pojem komunikácia sa na začiatku objavuje vo vedeckej literatúre. 20. storočie a veľmi rýchlo spolu so svojou všeobecnou vedeckou hodnotou (ako prostriedok komunikácie akýchkoľvek objektov) získa sociokultúrny význam spojený so špecifikami výmeny informácií v spoločnosti. Potrebné podmienky a konštrukčné prvky sú prítomne medzi predmetmi komunikácie, kanály prenosu informácií, ako aj pravidlá implementácie komunikácie (semiotické, etické). Rôzne interpretácie zdôrazňujú jej podstatu buď ako súbor prostriedkov na prenos sociálnych informácií, ktoré tvoria základ pre formovanie a rozvoj informačnej spoločnosti, alebo ako spôsob dosiahnutia porozumenia jednej osoby druhou.

Komunikačne kanály alebo platformy dnes musia riešiť aj veľké podniky. Napríklad na organizáciu pracovného poriadku, na definovanie úloh zamestnancom. Tieto riešenia umožňujú komunikáciu na diaľku, čo v dnešnej dobe je bežná vec, pretože pojem homeoffice je dnes známy každému. Veľa spoločnosti potrebujú podobne nástroje na už nami spomenuté úlohy. My teda chceme predstaviť a porovnať niekoľko takých riešení.

### 1.1.4.1 Zoom

Prvá platforma ktorá nás zaujala je dnes najviac používaný Zoom. Zoom je program, ktorý organizuje online stretnutia používateľov, videokonferencie, skupinové rozhovory, zvukové hovory. Vyžaduje stiahnutie špeciálneho softvéru do smartfónu, počítača alebo tabletu. Zoom sa stal obľúbeným serverom pre videokonferencie, stretnutia, diskusie, lekcie, prednášky, webové semináre a neformálnu komunikáciu.

Možnosti ZOOM

* organizácia video alebo audio komunikácie;
* zdieľanie súborov;
* kolektívny rozhovor;
* výmena textových a grafických správ;
* úložisko dát;
* posielanie pozvánok na online konferenciu;

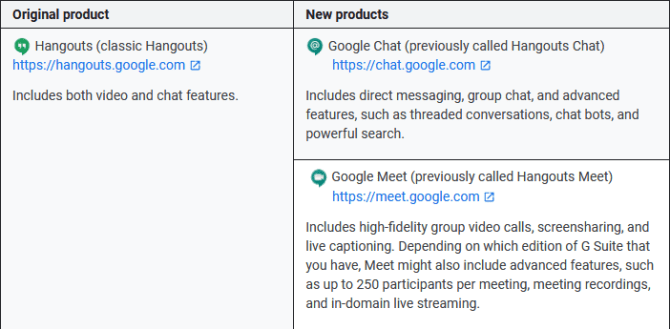
Zoom organizuje online stretnutia v niekoľkých formátoch. Konferencie - jedná sa o udalosti s interaktívnou súčasťou, určené pre mnohých účastníkov. Používatelia zdieľajú pracovnú plochu, zapínajú kameru a zvuk, sledujú zoznam účastníkov. Všetky konferencie sú rozdelené na okamžité a plánovane. Okamžite používajú sa na dôležité stretnutia, diskusie o detailoch projektu alebo iné urgentné záležitosti. Organizátor vytvorí konferenciu, odošle pozvánku používateľom, ktorí sa k nej pripoja pomocou odkazu. Plánované, ide o konferencie, ktorých dátum a čas sú stanovené vopred. V tomto formáte sa konajú lekcie, webové semináre, prezentácie atď. Účastníci konferencie poznajú tému vopred. Dali formát sú webináre (web semináre). V tomto formáte účinkuje niekoľko účastníkov vysielania. Na webinári používatelia sledujú, počúvajú a vymieňajú si informácie v zabudovanom chate. Zoom je multifunkčná platforma vhodná pre rôzne obchodné segmenty. Používa sa na profesionálne, vzdelávacie a osobné účely, má širokú funkčnosť a flexibilné nastavenie, čo pomáha používateľom pracovať s programom za výhodných podmienok. (NIKOLAI BOLSHAKOV, 2020)

### 1.1.4.2 Google Meet

Google Meet je služba videokonferencií od spoločnosti Google. Je to skvelé riešenie pre stretnutie jednotlivcov aj firiem pri audio a video hovoroch. Narodil sa v službe Google Hangouts, ale má niekoľko jedinečných funkcií. Spočiatku bol k dispozícii iba pre firemných zákazníkov, ale teraz môže každý používať Google Meet zadarmo. Google Meet je k dispozícii online cez webový prehliadač a tiež na telefónoch a tabletoch s Androidom a iOS.

Google Meet je primárne určený na videokonferencie. Fotoaparát a mikrofón však sa da zapnúť nezávisle, takže ich vieme podľa potreby použiť iba na zvukové hovory. Jednou z najlepších výhod služby Google Meet je, že nemusíme inštalovať žiadny softvér na plochu. Každý účastník hovoru (hostiteľ aj účastníci) musia použiť iba moderný webový prehliadač.

Google Meet vs Google Hangouts vs Google Chat



Google je známy tým, že má niekoľko chatovacích služieb, ktoré zjavne robia to isté. Google Hangouts je platforma pre video a správy. Google Hangouts sa rozdelil na Hangouts Meet a Hangouts Chat, ktoré sa neskôr premenovali na Google Meet a Google Chat. Ak chceme zorganizovať videokonferenciu pre viac ľudí, použijeme Google Meet. Ak chceme s niekým viesť individuálny rozhovor, formou video hovoru alebo textovej správy, použijeme službu Google Hangouts alebo Google Chat. Google Meet sa všemožne snaží odfiltrovať akýkoľvek šum na pozadí, ktorý nie je reč. Má tiež funkciu podpisu v reálnom čase, ktorá automaticky podpisuje to, čo ľudia hovoria - je to celkom spoľahlivé a skvelé. Môžeme tiež zdieľať svoju obrazovku s každým, kto je pripojený na hovor. Môže to byť konkrétne okno alebo celá pracovná plocha a funguje to aj na mobilných zariadeniach. Nakoniec spoločnosť Google tvrdí, že „videokonferencie sú šifrované pri prenose“ a že „náš bezpečnostný balík sa neustále aktualizuje, aby sa zvýšila ochrana“. Firemní používatelia dostanú ochranu na podnikovej úrovni, ale pre priemerného používateľa budú postačovať aj obyčajne nastavenia súkromia. (ISTOKS KLUB, 2020)

### 1.1.4.3 Microsoft Teams

Ešte jedným pre nás zaujímavým riešením je Microsoft Teams. Je to podniková platforma, ktorá spája v pracovnom prostredí čet, schôdzy, poznámky a prílohy. Je to tzv. centrum tímovej práce postavené na komunikácii, ktorá spája kolegov a partnerov a zahŕňa nástroje, ktoré tímy potrebujú na produktívnu spoluprácu. Tato platforma je nový priestor pre podnikovú komunikáciu. Integrované cloudové riešenie z Microsoft Office 365 zjednoduší našu IT infraštruktúru a zlepší zabezpečenie dát. Pomocou Microsoft Teams vieme vyriešiť množstvo problémov, ktoré trápia nás, náš tím a celú spoločnosť ako celok. Microsoft Teams vytvára digitálny pracovný priestor na základe týchto komponentov:

* **Čet pre dnešné tímy**. Microsoft Teams poskytuje modernú konverzáciu pre tímy - každý účastník projektu bude zapojený do konverzácie. Dialógy sú viditeľné pre celý tím, ale samozrejme existuje možnosť aj súkromných.
* **Centrum spolupráce**. Microsoft Teams je skutočným centrom spolupráce. Word, Excel, PowerPoint, SharePoint, OneNote, Planner, Power BI a Delve sú súčasťou Microsoft Teams. Teraz nie je potrebné strácať čas a úsilie hľadaním potrebných informácií a nástrojov – oni sú vždy po ruke. Microsoft Teams aplikácia ktorá pomáha ľuďom prechádzať z jednej úlohy na druhú, a to pomocou rôznych nástrojov a komunikácie so všetkými účastníkmi projektu.
* **Prispôsobivosť pre každý tím**. Nakoľko všetky tímy sú jedinečné, k dispozícii sú široké nastavenia a rozsiahla rozšíriteľnosť celého pracovného priestoru, otvorené verejné API. Napríklad karty poskytujú rýchly prístup k často používaným dokumentom a cloudovým službám. Microsoft Teams používa na zdieľanie model Connector, ktorý poskytuje oznámenia a aktualizácie zo služieb tretích strán, ako sú Twitter alebo GitHub. Okrem toho je poskytnutá úplná podpora pre Microsoft Bot Framework, ktorá prináša inteligentné služby do tímu. https://bizzapps.ru/p/microsoft-teams/

Hlavne dôvody prečo je dobre používať Microsoft Teams dnes:

* Rýchla komunikácia prostredníctvom video hovorov, sprav
* Možnosť odoslania a úpravy súborov v režime online
* Plánovať ďalšie úlohy a stretnutia pomocou kalendáru
* Možnosť vidieť status členov tímu
* Dostupnosť prístupu zo zariadení z rôznymi operačnými systémami (Windows, IOS, Android) (ROBOT IKS, ????)

Na zaver môžeme zhodnotiť že aplikácia Microsoft Teams je veľmi prakticky a pohodlný nastroj na komunikáciu, organizáciu tímovej prace, na stanovenie úloh a ich plánovanie.

## 1.2 Mobilne podnikové aplikácie

Podnikové aplikácie sú zamerané na pomoc organizáciám s ich obchodnými procesmi. Sú navrhnuté tak, aby integrovali všetky aspekty spoločnosti. Ako napríklad účtovníctvo, financie, ľudské zdroje, inventár a správa zdrojov.

Sú základom produktivity a efektívnosti interných procesov, dá sa povedať, že nám umožnia získať viac s menšou námahou. Aj keď je vytváranie takýchto aplikácií zložitým a časovo náročným procesom, kvalitné aplikácie tohto druhu vedú k vysokej úrovni zapojenia a lojality zákazníkov. Veľké a stredné podniky sa čoraz viac zaujímajú o vývoj bezpečných technických riešení, ktoré najlepšie vyhovujú ich potrebám. Dobre navrhnutá podniková aplikácia môže byť skvelým riešením, ktoré z dlhodobého hľadiska povedie k vyšším príjmom.

V dnešnej dobe nevieme si predstaviť majiteľa smartfónu, ktorý nepoužíva žiadnu z mobilných aplikácií. Dnešné mobilné aplikácie sú skutočne, pohodlné a rýchle riešenie, ako si objednať taxík alebo stôl v reštaurácii, zarezervovať hotel, kúpiť lístky do kina, poslať súbory, pripojiť sa na online míting atď. Tým však možnosti aplikácie nekončia.  
Mobilné aplikácie sa aktívne využívajú v podnikaní. Okrem toho riešia dva problémy naraz: optimalizujú procesy a zvyšujú firemnú kultúru. Hlavnými trendami vo vývoji podnikových aplikácií sú:

* **Personalizácia podnikovej aplikácie.** Personalizácia je typická pre softvérové ​​riešenia všeobecne: webové stránky aj mobilné aplikácie poznajú naše meno a zaujmi, gratulujú nám k narodeninám, ponúkajú zaujímavé články, videá a špeciálne ponuky.
* **Pohodlnosť použitia.** Základom mobilných aplikácií, vrátane firemných, je jednoduchosť a rýchlosť používania. Stačí len preťažiť funkčnosť - a zamestnanci budú potrebovať veľa času na to, aby sa naučili nové možnosti, pochopili, ako ich používať a či sú vôbec potrebné. A čas, ako vieme, sú peniaze. Našťastie je použiteľnosť trendová a podnikové aplikácie poskytujú iba funkcie, ktoré podnik skutočne potrebuje.  
  Nič navyše. Iba podnikanie.
* **Funkčnosť**. Len pred niekoľkými rokmi boli podnikové aplikácie iba doplňujúcom nástrojom, ktorý zamestnancom umožnil prístup k pracovným informáciám. A teraz podnikové aplikácie vychádzajú z tieňa. Ich funkcionalita sa rozširuje a teraz obsahuje GPS navigáciu, kontrolu transakcií, push notifikácie, sociálne siete a samozrejme odosielanie správ.
* **Ochrana podnikových a osobných údajov.** Mobilné podnikové aplikácie poskytujú ochranu na dvoch frontoch naraz. Prvým sú firemné údaje. Aplikácie umožňujú vytvárať čierne a biele zoznamy, konfigurovať prístup k informáciám, šifrovať ich a tiež požadovať autentifikáciu používateľa. A druhou časťou sú osobné údaje. Zamestnanci chcú vedieť, že spoločnosť nemá kontrolu nad ich osobnými údajmi. Preto je teraz trendom izolovaná prevádzka podnikových aplikácií, samostatná od iných aplikácií v mobile.
* **Komunikácia a socializácia.** Firemné mobilné aplikácie automatizujú veľa podnikových procesov, sledujú dopravne prostriedky spoločnosti, kontrolujú výsledky práce. To však nie je všetko.  
  Dôležitou funkciou podnikových aplikácií je komunikácia a socializácia. Veľa spoločností má zamestnancov ktorí vykonávajú prácu z domu formou home office, alebo zamestnanci z iných miest či dokonca krajín. Ako ich spojiť do priateľského tímu? Správne riešenie sú práve podnikové mobilné aplikácie. S ich pomocou si vieme vymieňať správy s kolegami, dozvedieť sa viac o firemných novinkách (úspešne implementované projekty, športové úspechy atď.). V podnikovej aplikácii je možne tiež poskytnúť vernostné programy pre zamestnancov: majstrovské kurzy, predplatné fitnescentra. To všetko zvýši motiváciu tímu. Firemné mobilné aplikácie sú teda užívateľsky prívetivé, intuitívne a ľahko použiteľné. Optimalizujú výkon spoločnosti a zvyšujú motiváciu zamestnancov. Pri rozhodnutí pre vytvorenie podnikovej aplikácie, je potrebne určiť rozsah úloh, ktoré by mala vykonávať, spísať všetky potrebné požiadavky. (PUNICAPP, ????)

## 1.3 Hlavne vývojové trendy vytvorenia dnešných biznis aplikácií

Vývoj podnikových aplikácií pre smartfóny a tablety je relatívne novým IT segmentom, ktorého potenciál ešte nie je využitý v celom svete. Mobilné aplikácie sa stávajú neoddeliteľnou súčasťou infraštruktúry moderného podniku. Po prvé, mobilné nástroje sa používajú na vzdialený prístup zamestnancov do podnikových systémov - môžu to byť ERP, CRM, BI aplikácie, skladové alebo logistické služby atď. Okrem zdrojov pre interných spotrebiteľov znamená mobilita aj vývoj aplikácií pre zákazníkov, pokiaľ ide o maloobchodné spoločnosti. Vlajkovou loďou tohto procesu sú bankové organizácie. Mobilné aplikácie svojim zákazníkom ponúkajú aj reštaurácie, obchody, taxislužby, fitnescentrá a ďalšie podniky pôsobiace v maloobchode. Mobilné služby možno označiť ako samostatnú oblasť pre interakciu s vládnymi úradmi, napríklad prostredníctvom mobilnej aplikácie si môžeme objednať termín, požiadať o vydanie dokladov, nahlásiť problém napríklad odfotiť dieru na ceste a poslať obrázok príslušnej mestskej službe.

Je potrebné poznamenať, že analytické agentúry dávajú konceptu „podnikovej mobility“ rôzny „obsah“. Odborníci zo spoločnosti Strategy Analytics naznačujú k tejto kategórií skutočné obchodné aplikácie také ako (CRM, ERP atď.), taktiež aj nástroje na spoluprácu a komunikáciu s kolegami (SMS, MMS, Messengeri, kalendáre, video konferenčné hovory). Analytici spoločnosti Gartner rozlišujú štyri prvky podnikovej mobility: podnikové aplikácie (Corporate Mobile Applications, CMA), platformy na vývoj týchto aplikácií (Mobile Enterprise Application Platform, MEAP), softvér na správu mobilného zabezpečenia (Mobile Enterprise Security, MES) a programy na správu podnikových mobilných zariadení (Enterprise Mobile Management, EMM). (PAVEL LEBEDEV, 2016)

V dnešnej dobe nie je možné si predstaviť prácu akéhokoľvek podniku bez IT systémov a aplikácií, ktoré nám umožnia pomôcť riadiť všetky procesy - od výroby až po logistiku. Mobilné aplikácie sa stavajú neoddeliteľnou súčasťou dnešnej podnikovej infraštruktúry – je to najefektívnejší spôsob optimalizácie a riadenia všetkých obchodných operácií.

Okrem toho vývoj podnikových mobilných aplikácií a integrácia aplikácií zjednoduší prácu zamestnancov, automatizuje a urýchli komunikáciu v rámci spoločnosti. Podnikové aplikácie sú rozdelené na interné a externé. Interné mobilné aplikácie používajú zamestnanci na vzdialený prístup k podnikovým systémom. Napríklad skladové a logistické služby.

Mobilné aplikácie pre zamestnancov sú to:

* Znížené prevádzkové náklady a prestoje.
* Schopnosť synchronizovať prácu jednotlivých tímových manažérov / pracovných tímov.
* Pomoc spravovať pracovné toky.

Pokiaľ sa jedná o spoločnosti pôsobiace v maloobchode, pojem „mobilita“ zahŕňa vývoj aplikácií pre zákazníkov. Lídrami v tomto procese sú bankové organizácie. Poďme sa pozrieť na najlepšie a najdôležitejšie trendy vývoja podnikových aplikácií vo vývoji podnikových aplikácií.

Personalizácia softvéru

Štruktúra dopytu sa mení spolu s rozvojom trhu s mobilnou infraštruktúrou podniku. Predtým neexistovali žiadne riešenia pre manažérov, ktoré by im umožňovali riadiť kľúčové obchodné procesy a boli veľmi žiadané, ale teraz nastal čas uspokojiť dopyt priemerných pracovníkov. Ľudia si zvykajú používať tablety a smartfóny na pracovné účely a chcú mať prístup k firemným službám v kaviarni, lietadle alebo aute. Zamestnávatelia majú tiež záujem zabezpečiť, aby boli zamestnanci vždy k dispozícii mimo kancelárie. Mobilné aplikácie teda radikálne menia spôsob fungovania spoločnosti. Používatelia navyše potrebujú mobilné aplikácie, ktoré ich spájajú s podnikovými prostriedkami. To významne zvyšuje produktivitu a uľahčuje komunikáciu medzi zamestnancami. Zamestnanci musia mať zároveň neobmedzený online prístup k informáciám a aplikáciám.

Pohodlne použitie

Rovnako ako bežné mobilné aplikácie, aj biznis aplikácie by mali byť užívateľsky príjemné a ľahko použiteľné. Vďaka tomu môžu spoločnosti znížiť čas a zdroje potrebné na školenie personálu. Teraz musia zamestnanci podnikov používať rôzne technologické nástroje a pracovať v rôznych oblastiach podnikania. To znamená, že proces používania softvéru by mal byť vhodný pre vysoko špecializovaných odborníkov.

Ak teda chceme vytvoriť aplikáciu pre podnik, musíme brať do úvahy zásadu, že aplikácia by mala nielen plniť svoje úlohy, ale aj byť ľahko použiteľná. Pretože mobilné aplikácie majú veľa funkcií, vývojári majú tendenciu implementovať všetky funkcie systému do mobilnej aplikácie. Musíme však neustále sledovať, či je aplikácia užívateľsky príjemná.

Funkčnosť

Spoločnosti zaoberajúce sa vývojom podnikových aplikácií predtým používali smartfóny hlavne ako ďalšie obrazovky, ktoré sa dali použiť na prístup k prevádzkovým informáciám bez toho, aby pokryli všetko, čo mohli získať z úplného rozhrania. Mobilné aplikácie majú veľmi širokú funkčnosť, prístup k navigácii a fotoaparátu mobilného zariadenia a v dôsledku toho majú väčší počet schopností a funkcií. V súčasnosti existuje trend smerom k plnému využívaniu funkcií inteligentných telefónov pre podnikanie. Je potrebné vytvoriť súpravu na vývoj softvéru na riešenie zložitých problémov: zhromažďovanie údajov zo senzorov smartfónov, optimalizácia životnosti batérie, zabezpečenie dôvernosti údajov atď. To vývojárom mobilných podnikových aplikácií umožní využívať celý potenciál smartfónu. Funkčnosť aplikácie závisí od cieľov konkrétneho oddelenia alebo spoločnosti ako celku. Napríklad mobilné aplikácie dokonale interagujú so všetkými integrovanými službami: sociálnymi sieťami, GPS navigáciou, notifikáciami. Tento nástroj je nevyhnutnosťou pre podniky, ktoré podnikajú v oblasti služieb.

Ochrana podnikových údajov a zabezpečenie mobilných aplikácií

Moderné podnikové aplikácie musia poskytovať izoláciu podnikových informácií a chrániť podnikové údaje pred modernými rizikami, ako sú napríklad:

* Neoprávnené použitie mobilných / cloudových aplikácií zamestnancami na vlastné účely.
* Únik podnikových údajov pomocou nespravovaných mobilných aplikácií / cloudu.
* Cielené útoky škodlivého softvéru na mobilné zariadenia.
* Neoprávnený prístup k podnikovým údajom.
* Strata alebo krádež zariadení.

Riešenia IT musia poskytovať konzistentnú úroveň zabezpečenia v celom podniku bez ohľadu na politiku zariadení. Môžeme vytvárať zoznamy povolených a nepovolených zariadení, upravovať zásady dodržiavania súladu a obmedzovať interné aplikácie. Zabrániť úniku údajov pomocou funkcií, ako je autentifikácia používateľa, šifrovanie údajov, kontrola zálohovania.

Zachovanie súkromia používateľov

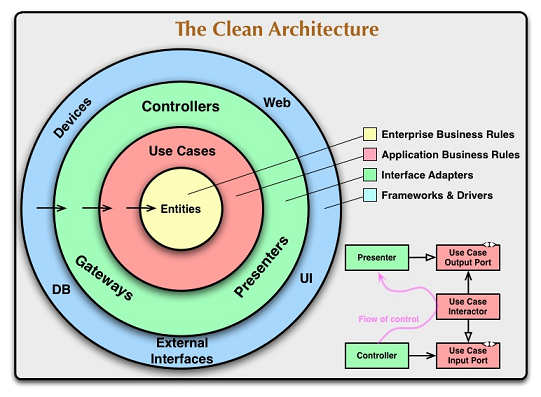
Vysokokvalitné mobilné aplikácie pre firmy musia chrániť zamestnancov. To znie zvláštne, pretože sa často kladie dôraz na prevenciu neoprávneného osobného použitia podnikových údajov. Ale je tiež pravda, že osobné informácie musia zostať súkromné. Zamestnanci a partneri by mali byť pokojní s vedomím, že spoločnosť nemá kontrolu nad ich osobnými údajmi. Tento trend je vhodný aj pre vývojárov, ktorí nechcú prevziať ďalšiu zodpovednosť a znášať náklady na správu osobných údajov. Dnešné mobilné obchodné prostredie si vyžaduje integrovaný prístup k bezpečnosti a správe vecí verejných. To je ideálne riešenie, ak používateľ nemá v pláne bezpečie. Vďaka pokročilej technológii obaľovania aplikácií, ktorá umožňuje podnikovým aplikáciám pracovať izolovane od jednotlivých aplikácií. Toto dáva zamestnancom výber zariadenia a priateľské užívateľské rozhranie. Spoločnosť si ponecháva kontrolu nad podnikovými aplikáciami a údajmi.

Zaver

Tu uvádzame niektoré z najpopulárnejších trendov vo vývoji web aplikácií pre podniky. Tieto aplikácie by mali byť priateľské pre podniky, škálovateľné a bohaté na informácie, užívateľsky prívetivé, prehľadné a ľahko použiteľné. Musia vyhovieť všetkým požiadavkám klienta a posunúť jeho podnikanie na vyššiu, vyššiu úroveň. Nakoniec je ich hlavným cieľom optimalizácia obchodných procesov, času a materiálov, zníženie nákladov a v dôsledku toho zvýšenie ziskov spoločnosti. (ARTUR AKOPAN, 2016)

# 2 Architektúra aplikácií

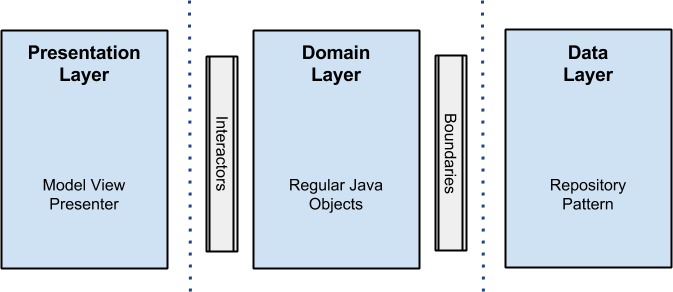
Vždy pri implementácii nejakého riešenia a presne aplikácií musí si zvoliť programátor architektúru ktorú použije vo svojej aplikácií. Nakoľko nás najviac zaujíma implementácia aplikácie pre Android, tak sa budeme zaoberať architektúrami ktoré sa používajú v takom prípade. Pred par rokmi najviac používanou architektúrou pre Androidove aplikácie typu klient-server bola architektúra ktorú navrhol Robert Martin, a je to „The Clean Architecture“.



Ako môžeme vidieť na obrázku, systém pozostáva z obchodných objektov, objektov na správu údajov a obchodnej logiky, prezentačnej vrstvy a rámcovej vrstvy na vysokej úrovni. Môžeme povedať, že vo všeobecnom prípade ide o rovnaké rozdelenie na prezentáciu údajov a logiku práce s nimi. Toto rozdelenie systému na vrstvy je veľmi logické a zrozumiteľné. A hoci neprináša nič nové. Hlavným je pravidlo závislostí - žiadna vnútorná vrstva by nemala vedieť nič o tej vonkajšej. To je to, čo nám umožňuje vybudovať nezávislú architektúru. Najkritickejšie z hľadiska testovania aplikácie je podniková logika alebo obchodné pravidlá, ktoré určujú podstatu aplikácie. Mali by byť primárne nezávislé od ostatných prvkov a zvlášť testované. Na dosiahnutie nezávislosti a možnosti testovania je aplikácia rozdelená na 3 kľúčové vrstvy:

* Dátová vrstva
* Vrstva podnikového myslenia
* Prezentačná vrstva

Zároveň aby zabezpečiť maximálnu nezávislosť týchto vrstiev, každá z nich používa vlastný dátový model, ktorý sa prevádza pri interakcii medzi vrstvami. Rozloženie týchto vrstiev je nasledovné:



Základná myšlienka všetkých architektúr zostáva - kód rozdeľujeme na moduly do niekoľkých vrstiev, medzi ktorými je aj vrstva obsahujúca logiku aplikácie a vrstva zodpovedná za prezentáciu údajov a prácu s UI.

Ale ako sa hovorí, dokonalosť nie je nijako obmedzená a takúto architektúru je možné a treba vylepšiť. Dnes existuje taký trend, keď sa všetka podniková logika prenáša na stranu servera. Tento trend je správny, pretože na strane servera je logika popísaná iba raz, a nie pre všetky mobilné platformy osobitne, a tiež preto, že logiku na strane servera je možné zmeniť oveľa rýchlejšie. (FANDROID, 2020) V takom to prípade skusmé sa pozrieť na viac zaujímavé pre nás architektúry, sú to tzv. MV architektúry.

Kód zložitej aplikácie, napísaný bez architektúr MV\*, je ťažké testovať, opakovane používať a udržiavať. Architektúry odstraňujú alebo oslabujú vzťah medzi View, Modelom a Controllerom, oddeľujú kód a zjednodušujú vývoj.

Predstavíme aplikáciu, ktorá má formu s rôznymi prvkami a widgetmi. Kód obsahuje popis logiky aj kód prvkov používateľského rozhrania, tiež môže obsahovať aj fragmenty na spracovanie údajov. Pre jednoduchú aplikáciu nie je udržanie tohto princípu programovania problémom. Vzťahy možno nájsť kedykoľvek a zmeny je možné vykonať bez narušenia celistvosti. Keď aplikácia sa stáva zložitejšou, vtedy sa stáva problém s údržbou neštandardného rozhrania, naprogramovaného bez architektúr.

Pri zmene kódu jedného komponentu, musíme urobiť zmeny v ostatných. Toto komplikuje údržbu aplikácie, automatické testovanie je takmer nemožné a opätovné použitie kódu je výrazne obmedzené.

Preto je pohodlné a správne používať architektúry, oni oddeľujú kód UI (View), logický kód (Presenter, Controller, ViewModel a ďalšie) a kód na spracovanie údajov (Model). Logiku môžeme ľahko zmeniť bez zmeny rozhrania alebo môžeme vykonať zmeny iba v postupoch spracovania údajov. Každú architektúru je možné testovať nezávisle a ďalej použiť aj v iných aplikáciách.

Najbežnejšie používané architektúry sú Model-View-Controller, Model-View-Presenter a Model-View-ViewModel. Skusmé ich popísať a pochopiť v čom sa líšia a kde sa používajú.

### 2.1 Popis architektúr

Model a View. Vo všetkých architektúrach, ktoré chceme popísať, sú dva opakujúce sa prvky. Líšia sa schopnosťami, ale sú v podstate rovnaké. View je vizuálne rozhranie (UI). Môže sa skladať z jednotlivých prvkov a widgetov. Príkladom View je kód na vytvorenie formulára v MFC a WinForms, html v ASP.NET, XAML vo WPF a Silverlight. Model sú údaje aplikácie, ktoré sa zobrazujú pomocou rozhrania, ako aj proces ich získavania a ukladania.

### 2.1.1 Architektúra MVC

Najstaršia architektúra, ktorá bola vyvinutá v roku 1979 pre vývoj aplikácií Smalltalk. V tom čase ešte neexistoval grafický obal Windows so štandardnými prvkami. Rozhranie bolo vykreslené ručne pomocou kódu. Už vtedy však bolo oddelenie zobrazovacieho kódu, logiky a údajov revolúciou vo vývoji.



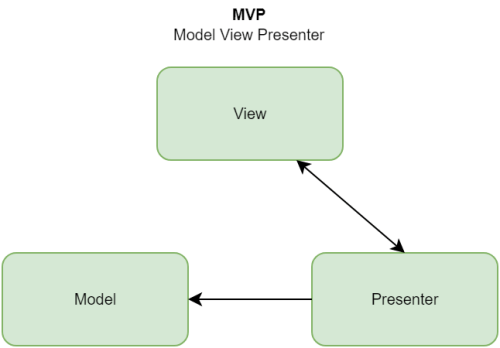
V tejto architektúre vidíme tri hlavné prvky:

* Model - údaje aplikácie s logikou ich prijímania a ukladania. Najčastejšie model pracuje s údajmi z databázy alebo s výsledkami webových služieb. Údaje sa zobrazia priamo na obrazovke alebo sa upravia.
* View - vizuálne rozhranie, kreslenie tlačidiel, štítkov, vstupné polia a ďalšie prvky formulára. Môže sledovať Model a zobrazovať údaje z neho.
* Controller - sleduje akcie používateľa (tlačidlá klávesnice alebo pohyby myši), rozhoduje, čo s nimi bude robiť, a aktualizuje Model a View.

Poďme si popísať, ako vzor MVC funguje. Controller ovláda činnosti používateľa - kliknutia myšou, stlačenie klávesnice alebo prichádzajúce požiadavky HTTP. Controller prenesie spracované zmeny do Modelu a nakreslí ich vo View (pasívny režim), alebo sa zmeny dostanú do Modelu priamo z View (aktívny režim). Hlavnou úlohou View, je zobraziť údaje z Modelu pomocou Controlleru.

### 2.1.2 Architektúra MVP

Vývoj vizuálneho programovania a widgetov eliminoval vykreslenie jednotlivých prvkov View, takže už nie je potrebná samostatná trieda Controller. Samotné prvky vedia, aké akcie s nimi užívateľ vykonáva. Stále je ale potrebné oddeliť aplikačnú logiku od údajov. Takto v architektúre sa urobila zmena - namiesto Controlleru sa objavil Presenter.



V porovnaní s MVC sa funkcia Model nezmenila, View teraz spracováva akcie používateľov sám (napríklad pomocou widgetov), ​​a ak táto akcia niečo zmení v logike rozhrania, potom sa odošle Presenterovi.

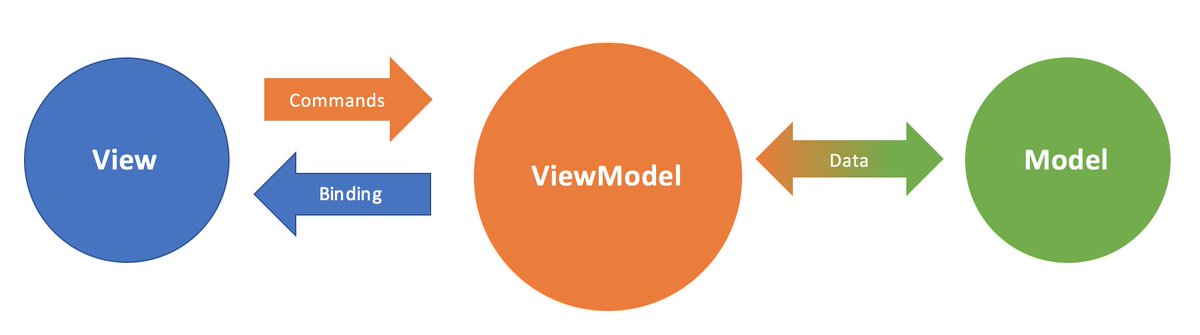
Hlavnou úlohou tejto architektúry je oddeliť View od Controlleru, aby bolo možné implementovať pripojiteľné View a umožniť ich nezávislé testovanie.

Presenter ako dirigent, je zodpovedný za synchrónnu činnosť Modelu a View. Ak dostane od View oznámenie o akcii vykonanej používateľom, aktualizuje Model a synchronizuje zmeny s View. Celá komunikácia prebieha prostredníctvom rozhrania, ktoré dáva ich oddelenie.

MVC má dve implementácie: Pasívne View, kde View nevie nič o Modeli, a Presenter je zodpovedný za prijímanie informácií z Modelu a aktualizáciu View, a Supervising Controller, kde View vie o Modeli a viaže dáta na samotné zobrazenie.

### 2.1.3 Architektúra MVVM

Kľúčovým rozdielom medzi touto architektúrou a ostatnými je prítomnosť databázového spojenia vo WPF a Silverlight.



Medzi ViewModel a View neexistuje priama komunikácia, deje sa to prostredníctvom príkazov (binding), pozostávajúcich z vlastností a metód. Týmto spôsobom môžeme spojiť akýkoľvek View a ViewModel, hlavná vec je mať potrebné vlastnosti. XAML binding tiež umožňuje spájať z View nielen údaje, ale aj akcie. Objekt nastavíme ako vlastnosť Model a deklaratívne ho naviažeme na zodpovedajúcu vlastnosť vo View. Vo výsledku dostaneme samostatný objekt, ktorý obsahuje údaje aj správanie, nezávisle od View. ViewModel je kombináciou Modelu a Controlleru.

Hlavnými výhodami MVVM sú ľahký dizajn rozhrania, nezávislé testovanie a redukcia kódu pre View.

### 2.2 Optimalizácia a výkon

Ak na oddelenie kódu View, Model a Controller použijeme vzory MV\*, potom v systémoch s vysokým zaťažením možno získať ďalšiu výhodu rozdelením týchto prvkov do rôznych výpočtových jednotiek.

View dávame na zariadenie klienta (laptop, PC, smartphone). Za zrýchlenie je zodpovedná technológia jednostránkovej aplikácie SPA. Na koncovom zariadení používateľa sa vykonávajú zložité výpočty, čím sa znižuje zaťaženie serverov typu backend.

Controller (Presenter, ViewModel) sa dáva na samostatný server typu backend, ktorý spracováva logiku.

Model je najobjemnejším a najproduktívnejším prvkom, preto vyžaduje server s rýchlym systémom ukladania údajov alebo lepšie so schopnosťou ukladať „horúce“ informácie do pamäte RAM.

Architektúry nie sú rigidné paradigmy, ktoré je potrebné používať, aby vytvoriť dokonalú organizáciu svojho kódu. Oni riešia veľmi dôležité problémy - oslabenie (elimináciu) väzieb medzi View, Modelom a Controllerom a zníženie zložitosti vývoja používateľských rozhraní. Implementácia tohto prístupu je možná, keď pochopíme podstatu vzťahu a nájsť možnosť ich eliminácie v každom konkrétnom projekte. (VLADIMIR CHAZOV, 2020)

# 3 Existujuce frameworky pre mikroservis

## 3.1 Definícia pojmu

Framework (kostra, štruktúra) - softvér, ktorý uľahčuje vývoj a integráciu rôznych komponentov veľkého softvérového projektu. Na rozdiel od knižníc, ktoré kombinujú množinu podprogramov s podobnou funkčnosťou, framework obsahuje veľké množstvo knižníc, ktoré sa líšia svojim účelom. Programátori si musia zvoliť, čo bude základom programovania. Jedným z najväčších rozhodnutí, ktoré musí programátor urobiť alebo programovací tím, je preto to, či je potrebne použiť framework alebo nie, a ak áno, tak ktorý. Pokúsime sa popísať výhody a nevýhody používania frameworku.

Tu sú niektoré zjavné výhody použitia frameworku, ktorý:

* Používa štandardné metódy riešenia problémov, ktoré znižujú alebo eliminujú zámenu kódu.
* Znižuje čas strávený rutinnými úlohami, ako je overovanie formulárov a zabezpečenie.
* Uľahčuje tímovú prácu dodržiavaním štandardov.
* Uľahčuje údržbu kódu pomocou bežnej architektúry a metód.
* Má aktívnu komunitu vývojárov, ktorí podporujú framework, všeobecné úlohy a nové funkcie.

Nevýhody

Existuje niekoľko negatívnych stránok pri používaní frameworku, ktoré by sme si mali všimnúť. Najskôr všetok kód, ktorý obsahuje framework, zaťaží server a to sa môže stať skutočným problémom. Našťastie sú k dispozícii možnosti ukladania do pamäte cache, ktoré znižujú dopad preťaženia, a pre podnikové aplikácie môžeme použiť natívny SQL na minimalizáciu času vykonania dotazu. Preto problém preťaženia servera by nám nemal brániť v používaní frameworku.

Ďalšia iná vec, keď pracujeme s frameworkom, zároveň pracujeme s obrovským množstvom kódu, ktorý sme nepísali a budeme musieť stráviť čas aby pochopiť ako to celé funguje. Niektoré časti kódu môžu byť dosť mätúce v závislosti od našej úrovne schopností a skúseností, takže by sme nemali očakávať, že všetkému porozumieme hneď.

Najjednoduchšie je teda pochopiť, že použitie frameworku nám pomôže vytvoriť štíhlejší a robustnejší projekt, teraz však prichádza najťažšie. Musíme sa rozhodnúť, ktorý framework použijeme.

Cyklus požiadavka-proces-odpoveď obsahuje mnoho prvkov, ktoré sú rovnaké pre všetky stránky. Preto vznikli takzvané frameworky - špecializované knižnice, ktoré určujú štruktúru programu. Je to ich rozdiel od bežných knižníc. Vďaka frameworkom sa môžeme sústrediť na logiku webu a nie na premýšľanie o základnej architektúre alebo podporných nástrojoch na kódovanie. Webové frameworky sú rozdelené do dvoch veľkých skupín, z ktorých jedna sa nazýva frameworky a druhá sú mikro-frameworky. Ale veľmi často pri vývoji malých projektov nie sú všetky funkcie frameworku jednoducho potrebné, stačia iba niektoré základné vlastnosti. V takom prípade prichádzajú na rad mikro-frameworky. Mikro-frameworky sú oveľa jednoduchšie a obsahujú iba minimálne nevyhnutné väzby pre pohodlnú prácu v architektúre HTTP - požiadavka-odpoveď. Sú ideálne na výučbu, pretože sú ľahko použiteľné.

Mikro-framework je rovnaká kostra/základ budúcej webovej aplikácie, ale je oveľa zjednodušená. To znamená, že práca s ním je jednoduchá a rýchla. Ktorý sa skladá iba z určitého softvérového jadra budúceho webu, skriptu alebo webového projektu. Hlavným cieľom mikro-frameworkov je jednoduché rozšíriteľné jadro. Neexistuje žiadna abstraktná databázová vrstva, žiadne overovanie formulárov ani nič, čo by už bolo implementované v rôznych knižniciach tretích strán, ktoré môžeme používať pri vývoje. Mikro-framework môžeme kedykoľvek sami rozšíriť a pridať potrebné funkcie.

Mikro-framework je aplikačný „rámec/základ“, ktorý obsahuje určitý druh smerovačov a typické úlohy, ktoré už boli vyriešené. Kľúčová časť slova tu je „mikro“. Veľmi ľahký, veľmi malý a veľmi rýchly produkt. Vo vnútri sa spravidla nachádza smerovač, ktorý definuje adresu URL, základnú implementáciu MVC, modely, radiče, spracovanie požiadaviek HTTP, spracovanie chýb. Je to minimum, ktoré potrebujeme na rýchle spustenie aplikácie. Toto minimum sa volá mikro-frework. Hlavná jeho definícia – malý a rýchly. Nie je nič nadbytočné. Funguje veľmi rýchlo a pre vývojárov vyzerá veľmi dobre. Veľmi jednoducho sa tam da pridať komponenty tretích strán, ktoré sa nabaľujú podľa princípu konštruktéra. Da sa pridať do neho všetko potrebné a rýchlo sa to implementuje a spúšťa. Na vytvorenie aplikácii nie je potrebné veľa štúdia a ponorenia sa do kódu.

### Vlastnosti mikro-frameworkov

Nie je veľa kódu, ktorom sa musíme vyznať. Z tohto dôvodu je jednoduchšie ladiť: samotný framework, ako aj aplikácie v ňom napísané. Avšak vývoj si samozrejme vyžaduje vyššiu kvalifikáciu, pretože veľa sa musí implementovať samostatne alebo samostatne integrovať do tohto mikro-frameworku. Existuje taká vlastnosť, že veľa mikro-frameworkov vyžaduje konkrétnu verziu PHP. Spravidla je to viac ako 5.3, pretože sa tam aktívne používa generovanie kódu, menné priestory na oddeľovanie komponentov.

URL smerovanie. Pekné adresy URL sú teraz podľa môjho názoru chceme vidieť všetci. Kopa parametrov GET v reťazci s adresou je dnes už neprehľadne. Každý chce vidieť malé, dobré a priateľské adresy URL. Také funkcie ako GET, POST, cookies, základné komponenty MVC, chceme to všetko vidieť v objektoch a pohodlne s nimi pracovať. Spravidla mikro-framework poskytuje základné implementácie modelov, základné implementácie radičov. Čo sa týka schémy ukladania do pamäte cache. Niektoré frameworky to nepodporujú a niektoré už majú niečo, čo dokáže ukladať do pamäte cache.

Prečo sú mikro-frameworky dobré. Da sa ich rozšíriť komponentami tretích strán. Navyše prakticky akékoľvek komponenty tretích strán sa dajú pridať. Napríklad vlastný ORM, TPL Engine, sieťovú komunikáciu, pripojenie API pre platobné systémy, REST/SOAP, pripojenie vlastnej pamäte cache. Je to veľmi jednoduché a prehľadne.

## 3.2 Lumen

Lumen je „mikro-framework“ postavený pomocou komponentov Laravelu a je oficiálnym mikro-frameworkom od spoločnosti Laravel. Lumen je navrhnutý tak, aby bol rýchly a je jedným z najrýchlejších mikro-frameworkov, dokonca podstatne rýchlejších ako podobný Silex. Avšak na rozdiel od mnohých iných mikro-frameworkov Lumen umožňuje využívať funkcie Laravelu, ako je smerovanie, vkladanie závislostí, Eloquent ORM, migrácie, fronty a dokonca aj plánované príkazy. Laravel je už rýchly a výkonný, ale Lumen odstránil mnoho doplnkov, ktoré Laravel poskytuje, aby udržať každú možnú milisekundu počas chodu vašej aplikácie. Neuveriteľná rýchlosť Lumenu v kombinácii s pohodlím a možnosťami Laravela vám dáva „najlepší“ svetový mikro-framework, s ktorým je radosť pracovať.

Lumen je navrhnutý tak, aby vytváral neuveriteľne rýchle mikroslužby a API. Napríklad ak časť nejakej aplikácie cez Laravel potrebuje oveľa vyšší prenos ako celá aplikácia, tak lepšie bude vytvoriť túto časť ako malú samostatnú aplikáciu pomocou Lumen. Znížením zaťaženia hlavnej aplikácie Laravel môžeme znížiť náklady na servery, pretože aplikácia postavená na Lumen nevyžaduje toľko zdrojov ako plnohodnotná aplikácia Laravel. Aplikácia Lumen samozrejme môže zaradiť úlohy do našej hlavnej aplikácie Laravel. Laravel a Lumen sú vytvorené nato aby vedeli spolupracovať ako jeden tím, tato spolupráca umožňuje vytvárať výkonné aplikácie riadené mikroslužbami. Lumen je vhodný aj na vytváranie rýchlych rozhraní API JSON, pretože také aplikácie zvyčajne nevyžadujú také funkcie, ako sú HTTP relácie, súbory cookie a šablónové nástroje.

Lumen nie je taký konfigurovateľný ako Laravel. Napríklad nie je možné zabrániť načítaniu frameworku, aby sa zmenila jeho konštrukcia. Na rozdiel od Laravelu tiež Lumen nie je navrhnutý na používanie balíkov pre Laravel, ako sú napríklad ladiace pruhy, systémy CMS atď. Lumen navyše nepoužíva smerovacie komponenty Symfony. Namiesto toho sa pre najlepší výkon používa nikic/fast-route. Ak potrebujeme smerovacie schopnosti Symfony, ako je smerovanie subdomény alebo voliteľné parametre, mali by sme použiť plnohodnotný framework Laravel. Ak sa rozhodneme použiť plnohodnotný Laravel, nemusíme sa báť slabého výkonu našej aplikácie. Laravel sa používa v mnohých veľkých podnikových aplikáciách, ktoré vybavujú až 15 000 000 požiadaviek za deň.

Lumen zahŕňa mnoho funkcií plnohodnotného frameworku Laravel:

* Šablóny Blade
* Ukladanie do vyrovnávacej pamäte(kešovanie)
* Príkazový plánovač(Command Scheduler)
* Kontroleri
* Eloquent ORM
* Správa chýb
* Abstrakcia databázy(Database Abstraction)
* Injekcia závislostí(Dependency Injection)
* Protokolovanie
* Fronty

Vďaka svojmu jedinečnému procesu načítania je Lumen schopný poskytnúť robustnú sadu funkcií a zároveň poskytnúť neuveriteľný výkon, čo je ideálne pre mikroslužby PHP.

## 3.3 HLEB

Mikro-framework sa líši od bežného frameworku tým, že umožňuje spustiť plnohodnotný produkt s minimálnymi časovými nákladmi a výzvami na dokumentáciu k frameworku, je ľahký, jednoduchý a rýchly. Zároveň rieši typické úlohy, ako je smerovanie (routing), presun akcií na radiče, podpora modelov, vo všeobecnosti základná implementácia MVC. To je minimum, ktoré potrebujeme na rýchle spustenie aplikácie.

Pre štandardné stránky existujú rôzne typy CMS, ale na implementáciu neštandardného nápadu sú zvyčajne potrebné flexibilné nástroje, ktoré umožňujú ich kombináciu. Na základe kombinácie medzi sebou aj fungujú moderne frameworky, čo nám umožňuje pripojiť svoje vlastné ďalšie moduly alebo moduly od vývojárov tretích strán. V mikro-frameworku sa dodáva „základ“, do ktorého sa pridáva kód projektu, ktorý pridáva programátor v rámci preddefinovanej štruktúry. Pretože vo väčšine prípadov jeden programátor pracuje na odľahčenej verzii frameworku s predponou „mikro“, implementuje úlohy podľa svojej vlastnej potreby štruktúrovania a ďalšej podpory, preto existuje menej požiadaviek na akékoľvek dodržiavanie pravidiel, aspoň v tomto prípade je tiež minimálne.

PHP Mikro-framework HLEB spĺňa tieto požiadavky, ale aby projekt na ňom založený plne odhalil svoj potenciál a bol tiež dostatočne chránený, je potrebné si naštudovať malý, ale nevyhnutný návod na použitie. Príručka sa skladá z nasledujúcich hlavných bodov.

Štruktúra projektu **-** Táto časť pojednáva o umiestnení pracovných priečinkov a súborov projektu. Tieto súbory sú všeobecne potrebné na komunikáciu s jadrom frameworku a neodporúča sa ich meniť mimo pokynov uvedených v tejto dokumentácii. Zvyšok, samotný projekt, jeho škálovanie a pripojenie knižníc tretích strán, je vývojárovi úplne k dispozícii.Smerovanie **-** Princíp smerovača je spracovávať prichádzajúce požiadavky a poskytovať pohodlný a moderný vzhľad adries URL, ktoré navyše vyzerajú dosť atraktívne. Po definovaní adresy nasleduje ďalšia vhodná reakcia aplikácie so zobrazením (alebo jej nedostatkom) údajov, ako aj s ďalšími priradenými akciami. To všetko je konfigurované vo forme diagramu trasy a vytvára mapu smerovania, podľa ktorej funguje celá stránka.

Radiče **-** Smerovač smeruje vykonávanie programu od určenia prichádzajúcej adresy k príslušnému radiču uvedenému v smerovacej mape, čo je v tomto prípade trieda PHP so špecifickou akciou a údajmi vrátenými používateľovi. Ak nie sú potrebné predbežné výpočty (napríklad kontrola autorizácie), môžeme určiť údaje vrátené (zo súboru alebo reťazca) v smerovači bez účasti radiču.

Konštruktér stránok **-** Konštruktér stránky je metóda zostavovania požadovanej stránky z rôznych súborov do jedného, ​​ktorá sa poskytuje používateľovi. Na spojenie rôznych častí stránky sa zvyčajne používa postupné vkladanie jej komponentov definovaných v týchto komponentoch do seba. Výhodou konštruktéra je, že jednotlivé súbory ako časti stránok sú navzájom nezávislé a ich mapovanie je priradené v smerovači s definíciou pomenovaných množín vrátane použitia montážnych množín. Táto funkcia, teda samotný nástroj na tvorbu stránok, je voliteľná a je možné ju použiť podľa potreby.

Architektúra **-** Kód frameworku HLEB sa vykonáva v procedurálnom štýle aj pomocou OOP. Každá funkcia bola doplnená o výber výkonu, minimalizmu a jednoduchého použitia. Tento prístup je spôsobený skutočnosťou, že tento mikro-framework zostane mikro-framework a nebude potrebovať škálovanie. Môžeme povedať, že prešiel štrukturálnou modifikáciou. Toto nie je odporúčaný štýl vývoja projektu, ktorý je na ňom založený, v súčasnosti existuje veľa príručiek, ako napísať škálovateľný a udržiavateľný kód, pripravený na prepracovanie a implementáciu novej funkcionality v krátkom čase.

## 3.4 Slim

Slim je PHP mikro-framework, navrhnutý na rýchle písanie jednoduchých, ale výkonných webových aplikácií a rozhraní API. Slim framework je v podstate dispečer, ktorý prijíma požiadavku HTTP, volá funkciu spätného volania priradenú k tejto požiadavke a vracia HTTP odpoveď. Slim Framework je ideálny nástroj na vytváranie rozhraní API, ktoré využívajú, opätovne používajú alebo ponúkajú prístup k údajom. Je to tiež vynikajúci nástroj na rýchle prototypovanie, keď si nápad vyžaduje skorú implementáciu. Okrem toho pomocou mikro-frameworku Slim môžeme vytvárať plne funkčné webové aplikácie s používateľským rozhraním. Je veľmi rýchly, má malú kódovú základňu - práve to, čo potrebujeme, a nič iné. Odporúča sa nainštalovať tento mikro-framework pomocou správcu závislostí Composer. PHP mikro-framework Slim je špeciálne navrhnutý takým spôsobom, aby čo najviac zjednodušil interakciu s komponentmi tretích strán. Pomocou komponentov tretích strán môžeme rozšíriť funkčnosť aplikácie. Tieto komponenty nájdeme predovšetkým v známom úložisku balíkov PHP - Packagist. Slim sa tiež dobre hrá s ostatnými komponentmi PHP. Môžeme pridať ďalšie komponenty tretích strán, ako napríklad Slim-Csrf, Slim-HttpCache alebo Slim-Flash, ktoré sa spoliehajú na predvolenú funkčnosť Slim. Je tiež ľahké integrovať komponenty tretích strán, ktoré nájdeme v Packagist. Pred spustením Slim, musíme mať nainštalovaný webový server. Webový server musí byť nakonfigurovaný tak, aby odosielal všetky požiadavky do predného radiča, t.j. do index.php. Predný radič vytvára, konfiguruje a spúšťa sa aplikácia. Aplikácia obsahuje trasy (URL), ktoré zodpovedajú konkrétnym požiadavkám HTTP. Každá trasa potom spustí zodpovedajúci obslužný program, ktorý predstavuje funkciu spätného volania a vráti odpoveď HTTP. Pri vytváraní aplikácie PHP Slim musíme takmer vždy pracovať s objektmi Request a Response. Tieto objekty zapuzdrujú požiadavky HTTP prijaté serverom a odpovede nimi vrátené v entitách, s ktorými je vhodné pracovať prostredníctvom objektovo orientovaného štýlu. Každá trasa má svoju vlastnú funkciu spätného volania, ktorá sa vykoná, ak sa požadovaná adresa zhoduje s trasou. Každej funkcii spätného volania sa odovzdávajú objekty odpovede a požiadavky. Implementujú populárny štandard PSR 7. Po prijatí žiadosti funkcia musí vrátiť objekt Response, ktorý je v súlade s daným štandardom. Na prácu so Slim frameworkom potrebujeme verziu PHP 5.5 a vyššie a tiež musíme mať nakonfigurované pravidlá konfigurácie servera na presmerovanie všetkých požiadaviek na index.php.

# 4 Ciele záverečnej práce

Cieľom záverečnej práce je identifikovať a prezentovať možnosti využitia prepojenia mobilných aplikácií s mikro-frameworkami podporujúcimi komunikáciu prostredníctvom API.

Na základe prezentovaných riešení navrhnúť a vytvoriť frontendové riešenie v podobe mobilnej aplikácie, pre operačný systém Android, ktorá bude zabezpečovať evidovanie a sledovanie činností i výkonu zamestnancov. Dôraz bude kladený na jednoduchosť prostredia a pozitívny používateľský zážitok.

Súčasťou aplikácie budú reporty prezentované vo faktografickej i grafickej podobe.

Podciele:

* analýza existujúcich typov podnikových aplikácií v súčasnosti,
* popis existujúcich architektúr aplikácií,
* identifikácia možnosti prepojenia mikro-frameworkov a mobilných aplikácií. a ich kľúčových charakteristík,
* popis, porovnanie mikro-frameworkov a ich princíp fungovania,
* návrh vlastnej aplikácie evidovanie činnosti zamestnancov,
* vyhodnotenie prace.

# 5 Návrh aplikácie

Naša aplikácia bude slúžiť ako portál na evidenciu pracovnej činnosti zamestnancov univerzity. Pred prvým použitím aplikácií používateľ bude mat možnosť prihlásenia do systému z už vytvorenými údajmi, taktiež bude možnosť registrácie nového užívateľa v systéme. Pri registrácii užívateľ si bude musieť zadať svoje osobne údaje, osobne číslo zamestnanca, pracovný mail a vybrať svoje pracovisko z poskytnutého zoznamu. K dispozícii na hlavnej obrazovke bude používateľovi prehlaď medzi hlavnými kategóriami a to projekty, publikácie a pracovne výkazy. Pri zvolení jednej z kategórii používateľ bude mat k dispozícii ďalšiu podrobnejšiu štruktúru ktorá bude obsahovať potrebne jemu údaje napríklad zoznam zamestnancov, zoznam projektov, zoznam publikácii a tď. Používateľovi bude dovolene v samostatných sekciách pridávať alebo odoberať rôzne položky ktoré si ďalej podrobne popíšeme.

Súčasťou návrhu nasej aplikácie je aj definovanie obsahu každej jej obrazovky.

Funkčne požiadavky aplikácie sú nasledovne:

* Možnosť prihlásenia pod osobnými údajmi a registrácia nového užívateľa.
* Zobrazenie a prezeranie obsahu projektov a publikácii.
* Zobrazenie personálnych údajov prihláseného užívateľa.
* Možnosť vytvorenia pracovného výkazu zamestnanca.
* Evidencia dochádzky a pracovných úkonov zamestnanca.
* Možnosť pridania nových projektov a publikácii.

Nefunkčne požiadavky:

* Nutnosť pripojenia na sieť Internet.

## 5.1 UML návrh

## 5.2 Návrh prihlasovacej obrazovky

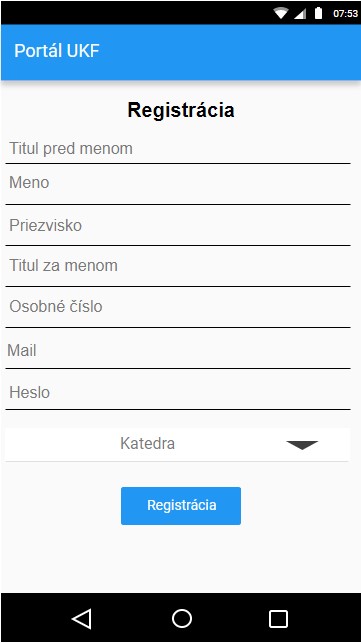
Tato obrazovka sa vždy otvorí pri spustenie aplikácií. Slúži na prihlásenie do aplikácie a presne do osobného úctu užívateľa. Tato aktivita obsahuje dve polia na zadanie e-mailu a hesla ktorými disponuje užívateľ. V prípade že používateľ je novy, vie si pomocou tlačidla „Registrácia“ založiť novy účet. Pomocou druhého tlačidla „Prihlásiť“ užívateľ sa prihlasuje po zadaní všetkých potrebných údajov.



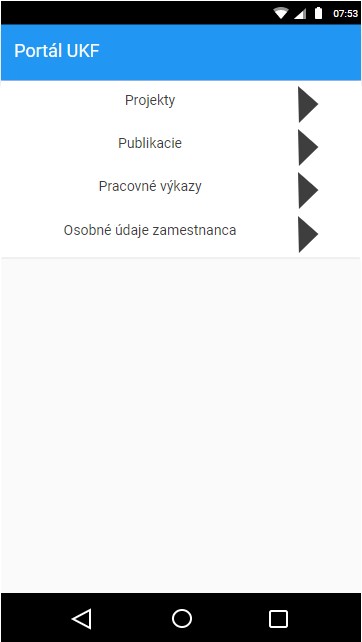
## 5.2.1 Proces autorizácie užívateľa

## 5.2.2 Proces registrácie užívateľa

V prípade že užívateľ aplikácie je novy, tak každý si ma právo založiť svoj osobný účet. Používateľ si musí vyplniť všetky požadovane údaje a to sú: tituly, meno, priezvisko, osobne číslo zamestnanca, pracovnú mailovú adresu a vytvoriť svoje osobne heslo. Je nutne tiež zvoliť z ponúknutého zoznamu aj katedru alebo iné pracovisko zamestnanca. Potvrdením tlačidla „Registrácia“ aplikácia dostane požiadavku na založenie nového úctu užívateľa, ktorú následne odošle na server a uloží nove údaje do databázy. Po úspešnej registrácii užívateľ sa vie prihlásiť do svojho osobného úctu.



## 5.3 Návrh domovskej obrazovky



## 5.4 Návrh obrazoviek so zoznamom

## 5.4.1 Zoznam fakúlt a jednotlivých pracovísk

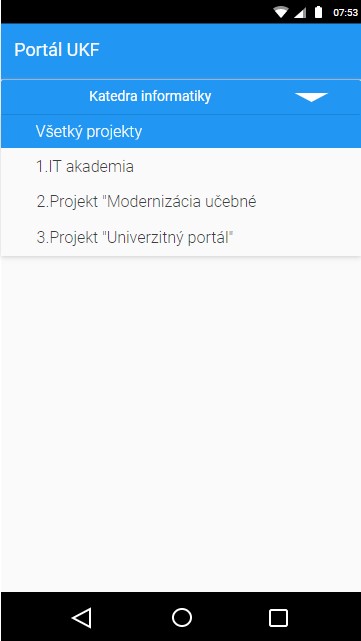
## 5.4.2 Zoznam katedier a iných pracovísk

## 5.4.3 Zoznam zamestnancov

## 5.4.4 Zoznam publikácií

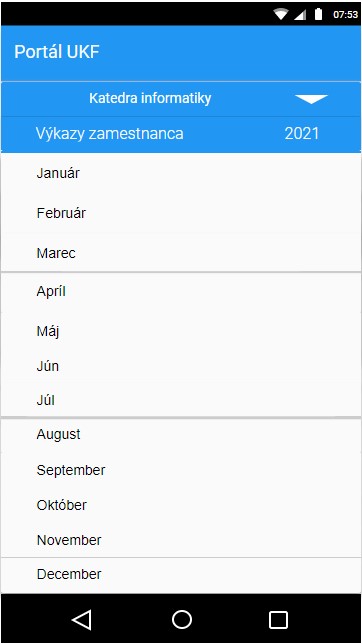
## 5.4.5 Zoznam projektov

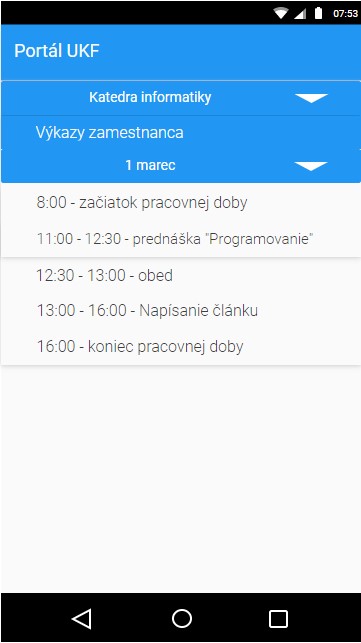
Každý používateľ bude mat prístup k zobrazeniu projektov v ktorých sa zúčastnil alebo do ktorých je zapojený na daný čas. Na prehlaď budú k dispozícii aj všetky projekty katedry alebo fakulty, súčasťou ktorých je zamestnanec.



## 5.4.6 Zoznam výkazov

V tejto kategórii si vie každý zamestnanec resp. používateľ aplikácie otvoriť svoj výkaz prace. Výkazy sú najprv zobrazovane a triedene podľa rokov a jednotlivých mesiacoch. Následne po zobrazení detailov pracovného mesiacu, používateľ ma k dispozícii na pridanie alebo úpravu každého pracovného dna. Zamestnanec pomocou poskytnutej funkcii vie pridať začiatok pracovného dna, koniec, prerušenie alebo ďalšie pracovne aktivity na ktorých strávil čas. Súčasťou každého riadku samostatného pracovného dna je dátum, čas konania pracovného dna od a do, a popis každej jednotlivej činnosti. Používateľ taktiež vie vymazať celu dochádzku za zvolený mesiac alebo odstrániť niektoré činnosti.





## 5.5 Endpointy

Endpointy je tzv. aplikačné programové rozhranie (API), ktoré umožňuje vzájomnú komunikáciu dvoch systémov. Endpoint je tzv. koncový bod. Jednoducho povedané, koncový bod je jeden koniec komunikačného kanála. Keď rozhranie API interaguje s iným systémom, kontaktné body tejto komunikácie sa považujú za koncové body. V prípade rozhraní API môže koncový bod obsahovať adresu URL servera alebo služby. Každý koncový bod je umiestnenie, z ktorého majú API prístup k zdrojom, ktoré potrebujú na vykonávanie svojej funkcie. API pracujú pomocou „požiadaviek“ a „odpovedí“. Keď API požaduje informácie z webovej aplikácie alebo webového servera, dostane odpoveď. Miesto, ktoré API odosielajú požiadavky a kde zdroj žije, sa nazýva koncový bod.

Rozhranie API označuje súbor protokolov a nástrojov, ktoré umožňujú interakciu medzi dvoma rôznymi aplikáciami. Zjednodušene povedané, je to technika, ktorá umožňuje predajcom tretích strán písať programy, ktoré sa dajú ľahko prepojiť. Na druhej strane je koncový bod miestom interakcie medzi aplikáciami. API označuje celú skupinu protokolov, ktorá umožňuje komunikáciu medzi dvoma systémami, zatiaľ čo koncovým bodom je adresa URL, ktorá umožňuje API získať prístup k prostriedkom na serveri.

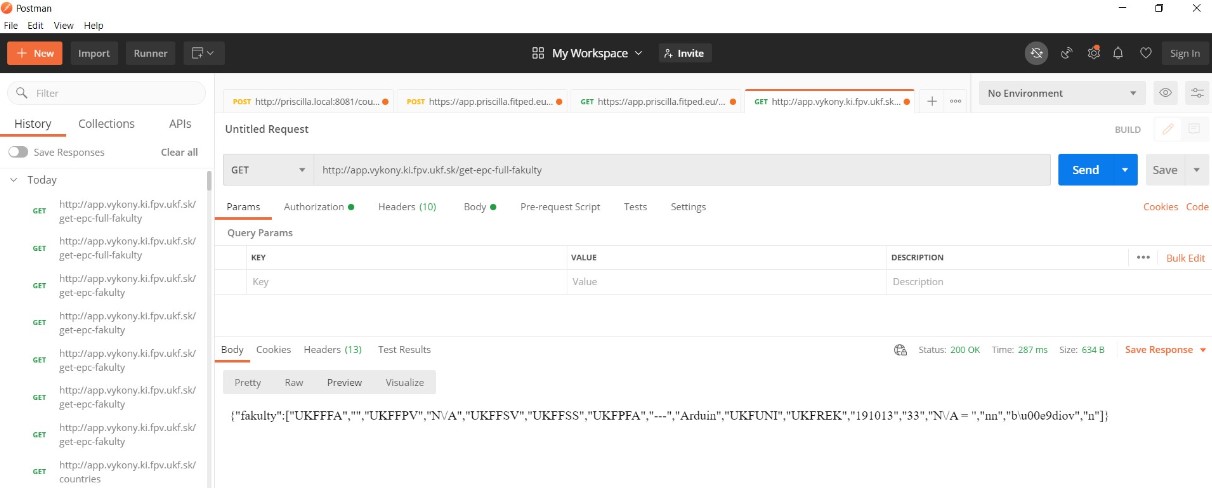
API v podstate poskytuje jazyk a zmluvu o tom, ako dva systémy interagujú. Každé API má dokumentáciu a špecifikácie, ktoré určujú, ako je možné prenášať informácie. Rovnako ako sa vykresľuje webová stránka, rozhrania API môžu na získanie informácií z webovej aplikácie alebo webového servera používať žiadosti HTTP. API sú zvyčajne kategorizované ako SOAP alebo REST a obe sa používajú na prístup k webovým službám. SOAP sa pri poskytovaní služieb zasielania správ spolieha iba na XML, zatiaľ čo REST ponúka ľahšiu metódu, ktorá na získanie alebo odoslanie informácií vo väčšine prípadov využíva adresy URL. REST používa na vykonávanie úloh štyri rôzne slovesá HTTP 1.1 (GET, POST, PUT a DELETE). Na rozdiel od protokolu SOAP nemusí REST na zabezpečenie odpovede používať XML. Vieme nájsť webové služby založené na REST, ktoré vydávajú údaje v súboroch Command Separated Value (CSV), JavaScript Object Notation (JSON) a Really Simple Syndication (RSS). Jedná sa o to, že požadovaný výstup môžeme získať vo forme, ktorá sa dá ľahko analyzovať v jazyku, ktorý potrebujeme pre našu aplikáciu. Po celom svete využívajú spoločnosti API na prenos dôležitých informácií, procesov, transakcií a tď. Využitie API sa bude len časom zvyšovať. Stručne povedané, výkon API závisí od jeho schopnosti efektívne komunikovať s koncovými bodmi API.

### 5.5.1 Backend rozhranie

Pre plnú funkčnosť nasej aplikácie potrebujeme taktiež aj backendove riešenie. Naše backend rozhranie sme aplikovali pomocou naprogramovaných endpointov. Naše endpointy využívajú hlavne metódy GET alebo POST. Pri POST požiadavke musíme uviesť aj dáta ktoré potrebujeme aby nás endpoint vrátil. Pre všetky hlavne úlohy ktoré vykonáva naša aplikácia sme naprogramovali endpointy, uvedieme niekoľko z nich.

* axios.get('/get-epc-full-fakulty', axios) – vráti skratky všetkých fakúlt alebo samostatných pracovísk univerzity
* axios.post('/get-epc-full-fakulty', data, axios) – vráti skratky všetkých katedier aj iných pracovísk pre fakultu
* axios.post('/get-zamestnanci-katedry, data, axios) – vráti zoznam zamestnancov katedry
* axios.post('/ get-projekty-zamestnanca, data, axios) – vráti zoznam všetkých projektov zamestnanca
* axios.post('/ get-epc-pracovisko, data, axios) – zobrazí všetky publikácie zvoleného pracoviska
* axios.post('/register', formData) – endpoint pre registráciu nového užívateľa

Toto sú príklady endpointov ktoré sme použili v nasej aplikácií. Na to aby sme vedeli overiť správnosť a funkčnosť nami napísaných endpointov sme použili aplikáciu Postman, ktorá nám zároveň aj vedela vrátiť výsledok na našu požiadavku a štruktúru, na to aby sme potom vedeli pri vývoji frontendového riešenia ako mame použiť dáta ktoré nám vrátil endpoint. Na obrázku môžeme vidieť výsledok zadanej nami požiadavky na vrátenie zoznamu fakúlt a samostatných pracovísk.



# Záver

# Zoznam BIBLIOGRAFICKÝCH ODKAZOV

PUNICAPP, (????). 5 трендов в развитии мобильных корпоративных приложений. [online] [cit. ????-??-??]. Dostupné na: <https://punicapp.com/blog/pages/614/5-trendov-v-razvitii-mobilnyh-korporativnyh-prilozhenij/>

NASTROYVSE, (????). SAP – ЧТО ЭТО ЗА ПРОГРАММА, ОСОБЕННОСТИ И ВОЗМОЖНОСТИ СОФТА. [online] [cit. ????-??-??]. Dostupné na: <https://nastroyvse.ru/programs/review/sap-chto-eto-za-programma.html>

ABDUL NIMERI, (2020). Что такое SAP?. [online] [cit. 2020-02-07]. Dostupné na: https://habr.com/ru/post/487418/

CLEVERENCE, (2020). SAP программа: что это такое простыми словами и как в ней работать – как расшифровывается программное обеспечение САП. [online] [cit. 2020-05-19]. Dostupné na: <https://www.cleverence.ru/articles/auto-busines/sap-programma-chto-eto-takoe-prostymi-slovami-i-kak-v-ney-rabotat-kak-rasshifrovyvaetsya-programmnoe/>

VALENTINA MANYAKHINA, ARTYOM ZOLOCHEVSKYва, (2016). Описание настройки и использования Moodle. [online] [cit. 2016]. Dostupné na: <https://docs.altlinux.org/ru-RU/archive/4.1/html-single/school-server/moodle/index.html>

RUSLAN YAKUPOV, (2018). Система электронного обучения и тестирования Moodle: обзор возможностей. [online] [cit. 2018-05-19]. Dostupné na: https://www.ispring.ru/elearning-insights/moodle

JEDNÉ CENTRUM VYSOKOŠKOLSKÉHO VZDELÁVANIA, (????). Система дистанционного обучения Moodle, что это такое и как ею пользоваться. [online] [cit. ????]. Dostupné na: <https://ecvdo.ru/states/sistema-distancionnogo-obucheniya-moodle-chto-eto-takoe-i-kak-eyu-polzovatsya>

ROBOT IKS, (????). Что такое и зачем нужен Microsoft Teams. Краткий обзор функционала. [online] [cit. ????]. Dostupné na: <https://www.robotx.ru/oblachnye-tehnologii/komandnaya-rabota/obzor-microsoft-teams/>

PAVEL LEBEDEV, (2016). Основные тренды рынка мобильных приложений для бизнеса: простота&функциональность. [online] [cit. 2016]. Dostupné na: https://www.cnews.ru/reviews/mobilnost\_v\_biznese\_2016/articles/osnovnye\_trendy\_rynka\_mobilnyh\_prilozhenij\_dlya\_biznesa\_prostotafunktsionalnost

ARTUR AKOPAN, (2016). 5 трендов в развитии корпоративных приложений. [online] [cit. 2016-10-28]. Dostupné na: https://itcrumbs.ru/5-trendov-v-razvitii-korporativnyh-prilozhenij\_18981

VLADIMIR CHAZOV, (2020). MV\*-паттерны в разработке веб-приложения. [online] [cit. 2020-09-16]. Dostupné na: https://simpleone.ru/blog/mv-patterny-v-razrabotke-veb-prilozheniya/

NIKOLAI BOLSHAKOV, (2020). Что такое Zoom: полный обзор возможностей и инструкция по настройке. [online] [cit. 2020-11-23]. Dostupné na: https://blog.calltouch.ru/chto-takoe-zoom-polnyj-obzor-vozmozhnostej-i-instrukcziya-po-nastrojke/

ISTOKS KLUB, (2020). Что такое Google Meet и как это работает?. [online] [cit. 2020-05-18]. Dostupné na: https://www.istocks.club/ что-такое-google-meet-и-как-это-работает/2020-05-18/

LEKTSII, (2020). Образовательные платформы. [online] [cit. 2020-05-07]. Dostupné na: https://lektsii.com/1-101899.html

FANDROID, (2020). Лекция 5 по архитектуре андроид приложений. Паттерн MVP. [online] [cit. 2020-11-09]. Dostupné na: https://www.fandroid.info/lecture-5-on-the-architecture-of-the-android-application-mvp-pattern/

# Zoznam príloh

# 