

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Interaktyvios pratybos

Interactive lecture

Kursinis darbas

Atliko: 3 kurso 2 grupės studentas
Marijus Siliūnas (parašas)

Darbo vadovas: Dr. Kristina Lapin (parašas)

Vilnius – 2016

TURINYS

ĮVADAS	2
1. STUDENTO IDENTIFIKAVIMAS	3
1.1. NFC technologija	3
1.2. Bluetooth technologija	3
1.3. NFC ar Bluetooth?	4
1.4. Mobilioji programėlė studentams	4
1.4.1. Programėlių kūrimas pagal operacinę sistemą	4
1.4.1.1. Android	4
1.4.1.2. iOS	4
1.4.1.3. Windows Phone	5
1.4.2. Viena programėlė - visoms operacinėms sistemoms	5
1.4.2.1. Hybrid	5
1.4.2.2. Native	5
1.4.2.3. Hybrid ar Native	5
1.4.3. Naudojimo scenarijai	6
1.4.3.1. Prisijungimas	6
1.4.3.2. Įrenginio pririšimas	6
1.4.3.3. Lankomumo pažymėjimas	6
1.5. Skaitytuvas - išmanusis telefonas	6
1.5.1. Skaitytuvo programėlės funkcijos	7
1.5.1.1. Prisijungimas	7
1.5.1.2. Paskaitos pradžia	7
1.5.1.3. Paskaitos pabaiga	7
2. SISTEMOS VALDYMAS	8
2.1. Aplikacijų programavimo sąsaja (API)	8
2.1.1. Saugumas	8
2.2. Administravimo aplinka	8
REZULTATAI IR IŠVADOS	9
SANTRUMPOS	10
PRIEDAI	10

Įvadas

Darbo tikslas

Šiuo darbu siekiama palengvinti darbą dėstytojams paskaitų metu žymint lankomumą. Sukurti sistemą, kurioje studentai skatinami aktyviai dalyvauti paskaitose apdovanojant sistemoje už kiekvieną veiksmą. Atsiskaitymų metu studentams bei dėstytojui aiškiai pateikti atsiskaitomą darbą. Sukurti sistemą, kuri leistų teikti kokybišką atsaką akivaizdinių studijų užsiėmimų metu, gerina studentų ir dėstytojų bendradarbiavimą.

Temos aktualumas

Statistika apie technologijas, dabartinė situacija

Siekiami rezultatai

Ištirti dėstytojų pasikartojantys procesai paskaitų metu ir nustatyta kuriuos iš jų galima automatizuoti. Sukurta sistema su automatizuotu studentų identifikavimo procesu bei studentų skaitinimo aktyviai dalyvauti paskaitoje mechanizmu.

1. Studento identifikavimas

Paskaitų metu yra žymimas studentų lankomumas. Šį darbą atlieka dėstytojas kiekvienos paskaitos metu. Studentai kviečiami vardu ir atsiliepę yra pažymimi kaip dalyvaujantys paskaitoje. Taip nereikalingai užimamas paskaitos laikas bei atsiranda žmogiškos klaidos tikimybė. Tai ypač svarbu, kuomet galutinis įvertinimas priklauso ir nuo lankomumo. Dėstytojams tai papildoma atsakomybė ir jie turi patys pasirūpinti priemone, kurioje galėtų vesti studentų lankomumo žurnalą, nes nėra bendros sistemos universitete. Reikalinga sistema, kurioje būtų suvedami dėstomi dalykai, dėstantys dėstytojai bei studentų grupės ir studentai. Sistemoje būtų vedamas lankomumo žurnalas.

Lankomumo žymėjimą galima visiškai automatizuoti. Taip būtų supaprastinamas dėstytojų darbas bei taupomas laikas. Lankomumo žymėjimą galima automatizuoti ne vienu būdu, tačiau pirmiausia reikia studentus identifikuoti.

1.1. NFC technologija

Vienas iš būdų yra Lietuvos Studento Pažymėjimo (toliau - LSP) panaudojimas studento identifikavimui. Naujo pavyzdžio LSP turi integruotą NFC (Trumpinys NFC yra angliško termino „Near Field Communication“ atitikmuo, apibūdinantis nedideliame nuotolyje veikiančią bevielio ryšio standartą, kuris užtikrina kokybišką, patikimą ir saugų ryšį tarp dviejų elektroninių įrenginių) technologiją. Jos pagalba galime nuskaityti LSP unikalų identifikavimo numerį ir pagal jį identifikuoti studentą. Identifikavus studentą sistemoje pažymimas jo dalyvavimas paskaitoje.

Visi studentai pradėję studijuoti įsigyja LSP todėl studentams toks procesas nesukeltų jokių nepatogumų. Norint įgyvendinti tokį identifikavimo procesą dar reikalinga integracija su duomenų baze, kurioje būtų susiejamas unikalus LSP identifikavimo numeris su LSP duomenimis. Gauti duomenys turėtų būti patikrinami ir su vidine universiteto sistema patikrinant, ar toks studentas tikrai egzistuoja, ar jis turi dalyvauti šioje paskaitoje ir t.t. Taip pat universiteto auditorijose turėtų būti įdiegti NFC skaitytuvai. Norint efektyviai atlikti žymėjimą turėtų būti įrengtas ne vienas, nes pažymėti galima tik po vieną studentą.

1.2. Bluetooth technologija

Kitas būdas galėtų būti panaudojant Bluetooth technologiją. Pati technologija nėra nauja, tačiau vis tobulinama. Beveik visi šiuo metu naudojami mobilieji telefonai turi integruotą šią technologiją. Mobiliuosius telefonus turi visi studentai. Kiekvienas įrenginys, kuriame yra integruota Bluetooth technologija, turi savo unikalų identifikavimo numerį, kurį galima nuskaityti. Kaip ir prieš tai pateiktu būdu - šį numerį reikėtų susieti su studento informacija. Privalumas prieš LSP skaitymą - nereikia integruoti trečiųjų šalių paslaugų. Identifikaciniai numeriai išsaugomi ir susiejami su studentu kuriamoje sistemoje.

Universiteto auditorijose taip pat turėtų būti įdiegti skaitytuvai, kurie nuskaitytų studentų mobiliųjų telefonų unikalius identifikacinius numerius. Tokio skaitytuvo tereikėtų vieno auditorijoje, nes mobiliųjų telefonų siunčiamus signalus Bluetooth ryšius galima nuskaityti visus vienu

metu. Nors studentai turėtų įjungti šią technologiją telefone, pats procesas vyktų žymiai efektyviau ir paprasčiau.

1.3. NFC ar Bluetooth?

Pateikti būdai turi ir privalumų, ir trūkumų. Visgi tinkamesnis būdas identifikuoti studentus yra pasitelkus Bluetooth technologiją, nes: nepriklausoma nuo trečiųjų šalių paslaugų (services); auditorijose reikia įrengti mažiau skaitytuvų ir / ar greitesnis identifikavimas (nesusidaro eilės prie skaitytuvų).

1.4. Mobilioji programėlė studentams

Toks proceso įgyvendinimas prasideda nuo programėlės sukūrimo išmaniesiems telefonams. Taip pat svarbu, kad programėlė ne tik padėtų identifikuoti studentą, tačiau būtų kuo paprastesnė (reikėtų atlikti kuo mažiau veiksmų) naudotojui.

Kiekvienas įrenginys turintis integruotą Bluetooth technologiją turi ir unikalų MAC (Media Access Control) adresą, dar vadinamą fiziniu adresu. Kadangi šis adresas kiekvienam įrenginiui yra unikalus, tai juo galime naudotis norint identifikuoti studentus. Norint susieti studento mobilųjį telefoną su jo paskyra sistemoje studentas įsidiegia programėlę. Programėlėje studentas prisijungia prie savo paskyros ir “pririša” mobilųjį telefoną. Programėlė nuskaityto telefono Bluetooth MAC adresą ir kartu su studento identifikavimo numeriu (gaunamas prisijungus prie paskyros) siunčia į sistemą. Sistema gautą MAC adresą priskiria tam studentui. Programėlės pagrindinė funkcija yra “pririšti” telefoną prie paskyros. Tuo pačiu galima pateikti studento tvarkaraštį, lankomumo ataskaitą.

Norint, kad sistema galėtų naudotis kuo daugiau studentų, programėlė turi būti palaikoma skirtingose įrenginiuose bei operacinėse sistemose. Pačios populiariausios išmaniųjų įrenginių operacinės sistemos yra Android ir iOS. Taip pat gana plačiai naudojama “Windows Phone” operacinė sistema.

1.4.1. Programėlių kūrimas pagal operacinę sistemą

1.4.1.1. Android

Tai yra populiariausia, apie 60 procentų rinkos užimanti, operacinė sistema. Ši platforma yra atviro kodo ir prižiūrima “Google” kompanijos. “Android” operacinė sistema diegiama į tokių kaip “LG”, “Sony” ir kitų gamintojų išmaniuosius telefonus. Šiai platformai programėlės kuriamos Java programavimo kalba. Programavimo aplinkai kūrėjai siūlo nemokamą įrankį “Android Studio” bei pamokų ciklą pirmajai programėlei sukurti.

1.4.1.2. iOS

Antra pagal populiarumą, apie 25 procentus rinkos turinti, operacinė sistema. Ši platforma yra palaikoma ir sukurta “Apple” korporacijos. Programėlės kuriamos “Objective-C” arba “Swift”

programavimo kalbomis.

1.4.1.3. Windows Phone

Trečia pagal populiarumą, apie 15 procentų išmaniųjų įrenginių rinkos užimanti, operacinė sistema. Tai yra "Microsoft" korporacijos produktas. Operacinė sistema diegiama "Nokia" korporacijos "Lumia" išmaniųjų telefonų linijoje. Programavimo kalba - C#.

1.4.2. Viena programėlė - visoms operacinėms sistemoms

Dažnai poreikis yra palaikyti kuo daugiau įrenginių. Ir šiuo atveju - reikalinga, kad naudojamasis sistema būtų kuo prieinamesnis. Būtent dėl šios priežasties buvo pradėtos kurti technologijos leidiančios rašyti vieną kodą vienai programėlei ir ji veiktų visuose palaikomose operacinėse sistemose. Kol kas nėra tiek išvystyta technologija, tačiau šiuo metu galime su nedidelėmis korekcijomis sukurti programėlę tinkamą populiariausioms išmaniųjų įrenginių operacinėms sistemoms.

1.4.2.1. Hybrid

HYBRID. Vienas iš labiausiai paplitusių karkasų yra "Adobe PhoneGap". Naudojantis šiuo karkasu galime programėlę kurti kaip tinklalapį, programuojant JavaScript programavimo kalba bei rašant HTML ir CSS kodą vartotojo sąsajai kurti. "Adobe PhoneGap" taip pat mums suteikia priėjimą prie įrenginio operacinės sistemos aplikacijų programavimo sąsajos (angl. API - Application Programming Interface). Tokiu būdu mes galime pasiekti išmaniojo įrenginio resursus, o šiuo atveju - mums reikalingą Bluetooth modulį.

1.4.2.2. Native

NATIVE. Kita platforma, skirta kurti programėles Android, iOS ir Windows platformoms, tai Xamarin. Naudojantis šia platforma galima kurti visoms trimis mobiliosioms operacinėms sistemoms "native" programėles. Šiuo atveju programėlės yra rašomos C# programavimo kalba, o programinė įranga, reikalinga surinkti bei vystyti programėles, yra nemokama. Vidutiniškai 75% programėlės kodo yra naudojama bendrai tarp operacinių sistemų. Platforma suteikia pilną priėjimą prie įrenginio operacinės sistemos aplikacijų programavimo sąsajos (visose operacinėse sistemose). Vartotojo sąsają galima kurti pagal operacinės sistemos standartus arba naudojant platformos kūrėjų įrankį "Xamarin.Forms".

1.4.2.3. Hybrid ar Native

Tiek vienas, tiek kitas programėlių kūrimo būdas turi ir plusų, ir minusų. Hybrid tipo programėles galima kurti greičiau ir paprasčiau, tačiau jų veikimo sparta bei vientisumas gali kelti nepatogumų naudotojams. Taip pat yra sudėtingiau išnaudoti išmaniojo įrenginio ir jo operacinės sistemos teikiamas funkcijas, pateikiamas per aplikacijų programavimo sąsają. Tai vis dėlto yra

geresnis variantas nei rašyti kiekvienai operacinei sistemai atskirą programėlę, laiko ir kainos atžvilgiu. Visgi kuriamos sistemos realizacijai reikia kuo geresnės integracijos su išmaniuoju įrenginiu. Todėl tinkamiausias būdas kurti programėlę sistemai yra naudojant Xamarin platformą. Ji leis sukurti intuityvesnę, vientesnę bei patikimesnę programėlę.

1.4.3. Naudojimo scenarijai

1.4.3.1. Prisijungimas

Įdiegus ir pirmą kartą programėlę paleidus naudotojui atidaromas prisijungimo langas (wireframe / design priede ar čia?). Studentas suveda prisijungimo vardą bei slaptažodį. Spaudžia prisijungti. Sistemai gražinus sėkmingą pranešimą, programėlė išsisaugo prisijungimo duomenis (arba grąžintą access token, priklausomai nuo sistemos įgyvendinimo). Kitą kartą paleidus programėlę prisijungimas vyksta automatiškai.

1.4.3.2. Įrenginio pririšimas

Prisijungus programėlėje, meniu juostoje iškviečiame context? meniu, pasirenkame "Susieti įrenginį". Programėlė surenka reikalingą informaciją apie įrenginį (Bluetooth MAC) ir siunčia užklausą į sistemos aplikacijų programavimo sąsają. Gražinama žinutė apie sėkmingą arba nesėkmingą įrenginio susiejimą su paskyra.

1.4.3.3. Lankomumo pažymėjimas

Studentas atėjęs į paskaitą įsijungia programėlę bei įrenginio Bluetooth ryšį. Spaudžia "Pažymėti". Programėlė ieško skaitytuvo ir nuskaito jo siunčiamą informaciją: paskaitos pavadinimą, dėstytojo vardą ir pavardę. Rodomas patvirtinimo langas su šia informacija. Studentas spaudžia "Čia pataikiau". Patvirtinimas siunčiamas į sistemą ir pažymimas studento dalyvavimas paskaitoje.

Alternatyvus scenarijus: Studentui tereikia įjungti Bluetooth ryšį ir būti susiejus įrenginį su paskyra. Skaitytuvas aptikęs įrenginį automatiškai pažymi studento dalyvavimą paskaitoje.

1.5. Skaitytuvas - išmanusis telefonas

Kitas svarbus komponentas Bluetooth identifikacijoje - skaitytuvas. Tai galėtų būti atskiras prietaisas arba tiesiog išmanusis telefonas. Reikalinga sukurti atskirą programėlę, kurią įsidiegtų dėstytojas į išmanųjį telefoną.

Paskaitos metu programėlė nuskaito įrenginius esančius patalpoje ir siunčia įrenginių sąrašą su jų MAC adresais į sistemą, kur gautas unikalūs identifikatoriai susiejamas su studento paskyra ir pažymimas jo dalyvavimas paskaitoje. Norint tai įgyvendinti, dėstytojas taip pat turi turėti savo paskyrą sistemoje. Paskyroje pridedamos paskaitos, priskiriamos grupės bei studentai.

Kaip ir programėlė skirta studentams, taip ir skirta dėstytojams turi būti kuo prieinamesnė. Todėl jos kūrimui taip pat pasirinkta Xamarin platforma.

1.5.1. Skaitytuvo programėlės funkcijos

1.5.1.1. Prisijungimas

Įdiegus ir pirmą kartą programėlę paleidus naudotojui atidaromas prisijungimo langas (wi-reframe / design priede ar čia?). Dėstytojas suveda prisijungimo vardą bei slaptažodį. Spaudžia prisijungti. Sistemai grąžinus sėkmingą pranešimą, programėlė išsaugo prisijungimo duomenis (arba grąžintą access token, priklausomai nuo sistemos įgyvendinimo). Kitą kartą paleidus programėlę prisijungimas vyksta automatiškai.

1.5.1.2. Paskaitos pradžia

Paleidus programėlę ir prisijungus (automatiškai arba suvedus prisijungimo duomenis) pagrindiniame lange spaudžia "Pradėti paskaitą". Atidaromas paskaitų sąrašas, rikiuotas pagal paskaitos laiką (nurodoma administravimo dalyje). Paspaudžia ant paskaitos, atsiranda patvirtinimo langas, spaudžia "Pradėti". Programėlė pradeda skleisti informaciją Bluetooth ryšiu. Rodomas paskaitos langas su pasižymėjusiais studentais.

Alternatyvus scenarijus: Pradėjus paskaitą periodiškai skanuojamas Bluetooth ryšys ir rasti įrenginiai siunčiami į sistemą.

1.5.1.3. Paskaitos pabaiga

Paskaitos lange spaudžia "Užbaigti paskaitą". Rodomas patvirtinimo pranešimas, spaudžia "Užbaigti". Programėlė nusiunčia užklausą į sistemą dėl paskaitos užbaigimo. Rodomas pranešimas su sėkmingo / nesėkmingo užbaigimo tekstu. Atidaromas pagrindinis langas.

2. Sistemos valdymas

Apžvelgėme, koku būdu ir kokiomis priemonėmis sistemos naudotojai galės naudotis sistema. Bet visa tai neveiks be pagrindinės sistemos dalies. Kaip pavadinti backend? API?

2.1. Aplikacijų programavimo sąsaja (API)

Visas bendravimas tarp klientų (mobiliųjų programėlių, web?) ir sistemos vyksta per aplikacijų programavimo sąsają.

2.1.1. Saugumas

Generuojami tokenai, klientuose nesaugomi prisijungimo duomenys. Payload su encryption? Pasiklausti Eligijaus apie šitą.

2.2. Administravimo aplinka

Rezultatai ir išvados

Rezultatų ir išvadų dalyje turi būti aiškiai išdėstomi pagrindiniai darbo rezultatai (kažkas išanalizuota, kažkas sukurta, kažkas įdiegta) ir pateikiamos išvados (daromi nagrinėtų problemų sprendimo metodų palyginimai, teikiamos rekomendacijos, akcentuojamos naujovės).

Santrumpos

Sąvokų apibrėžimai ir santrumpų sąrašas sudaromas tada, kai darbo tekste vartojami specialūs paaiškinimo reikalaujantys terminai ir rečiau sutinkamos santrumpos.