

VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

„Kinder2“ UML 4+1

Autoriai:

Justas Baniulis 2 kursas, 1 grupė

Konstantinas Bernotas 2 kursas, 1 grupė

Olegas Borisovas 2 kursas, 1 grupė

Rokas Petrauskas 2 kursas, 1 grupė

Vilnius, 2022

1. Kontekstas

1.1. Programos būsena prieš pakeitimus:

„CityPuzzle“ – žaidimas, skatinantis keliauti, atrasti bei tyrinėti įvairius geografinius, istorinius ir kitus kultūrinę reikšmę turinčius objektus. Artėjant ar tolstant nuo tokio objekto, matomas atitinkamas indikatorius, o atstumui mažėjant – po truputį atsiveria ir minėtojo objekto nuotrauka. Jei rasti finišą sunku – galima naudoti pagalbos mygtuką, laikinai parodantį visą objekto vaizdą.

1.2. Motyvacija pokyčiams:

Šios sistemos pelnui generuoti reikalingi klientai, kurių įstaigos yra įtraukiamos į programėlę kaip lankytinas objektas. Siekiant klientą išlaikyti, jam turi būti suteikiama informacija, apie tai, kiek lankytojų apsilankė jo įstaigoje dėl mūsų sistemos. Nusprendėme, kad šiai statistikai pateikti bei kliento funkcijoms (pridėti objektą, pašalinti objektą ir t.t.) reikalinga atskira aplinka.

Didelę svarbą turi ir rėmėjai – turizmo informaciniai centrai, regioniniai parkai ir t.t. Jų pagalba pelnas generuojamas nėra, tačiau jie suteikia informacijos apie kelius, nenumatytas situacijas jų teritorijose, artėjančius įvykius bei pridėda turizmo traukos taškus į sistemą. Tai padėtų subalansuoti remiamų ir neremiamų objektų santykį, kas atvertų galimybę įtraukti vis daugiau klientų – tai netiesioginis verslo augimo stimulus. Kad rėmėjų aktyvumas būtų kuo didesnis, jų darbas turi būti kuo lengvesnis – reikalinga paprasta, patogi aplinka, užtikrinanti visą reikalingą funkcionalumą.

1.3. Įgyvendinti pokyčiai:

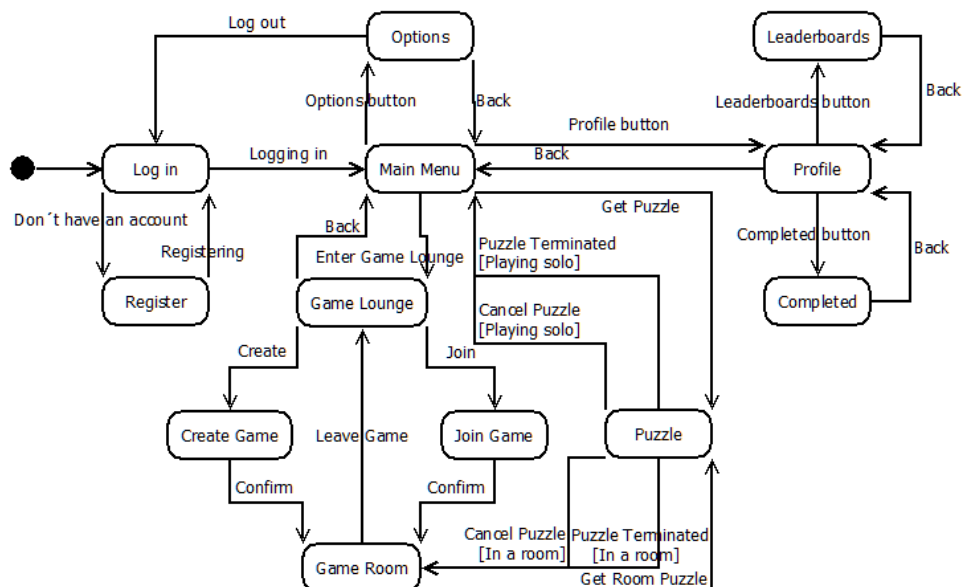
Sukurtas internetinis puslapis – aplinka, skirta tiek klientams, tiek rėmėjams, kadangi abiejų asmenų sistemos poreikiai sutampa. Šioje aplinkoje galima įtraukti turizmo traukos taškus, juos pašalinti ir redaguoti: įkelti bei ištrinti jų nuotraukas, aprašą. Leidžiama pateikti ir kitokią informaciją, kurią matys sistemos administratorius, pvz.: kelias laikinai uždarytas, regioniniame parke vyks renginys. Šioje aplinkoje taip pat prieinama informacija apie lankytojų, atvykusių dėl „CityPuzzle“ žaidimo, skaičių.

2. UML 4+1

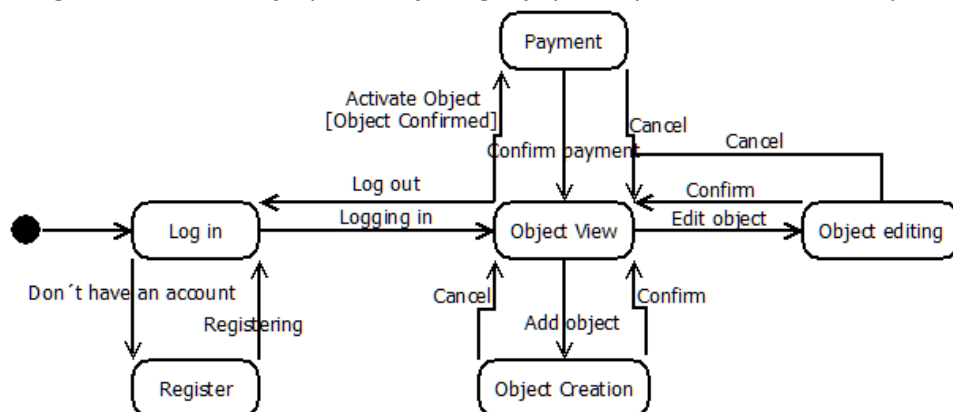
2.1. Logical view

„Logical view“ rodo kaip vartotojai, klientai ir partneriai judės pro skirtingas programų dalis.

2.1.1. Diagrama rodo vartotojo navigaciją pro CityPuzzle programėlę.

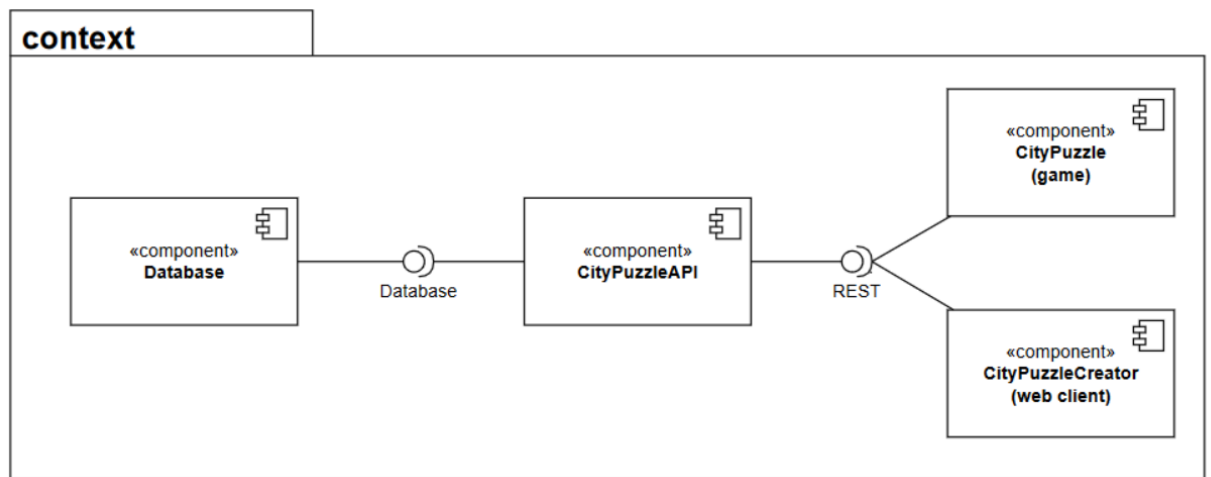


2.1.2. Diagrama rodo klientų ir partnerių navigaciją pro CityPuzzleCreator web aplikaciją.

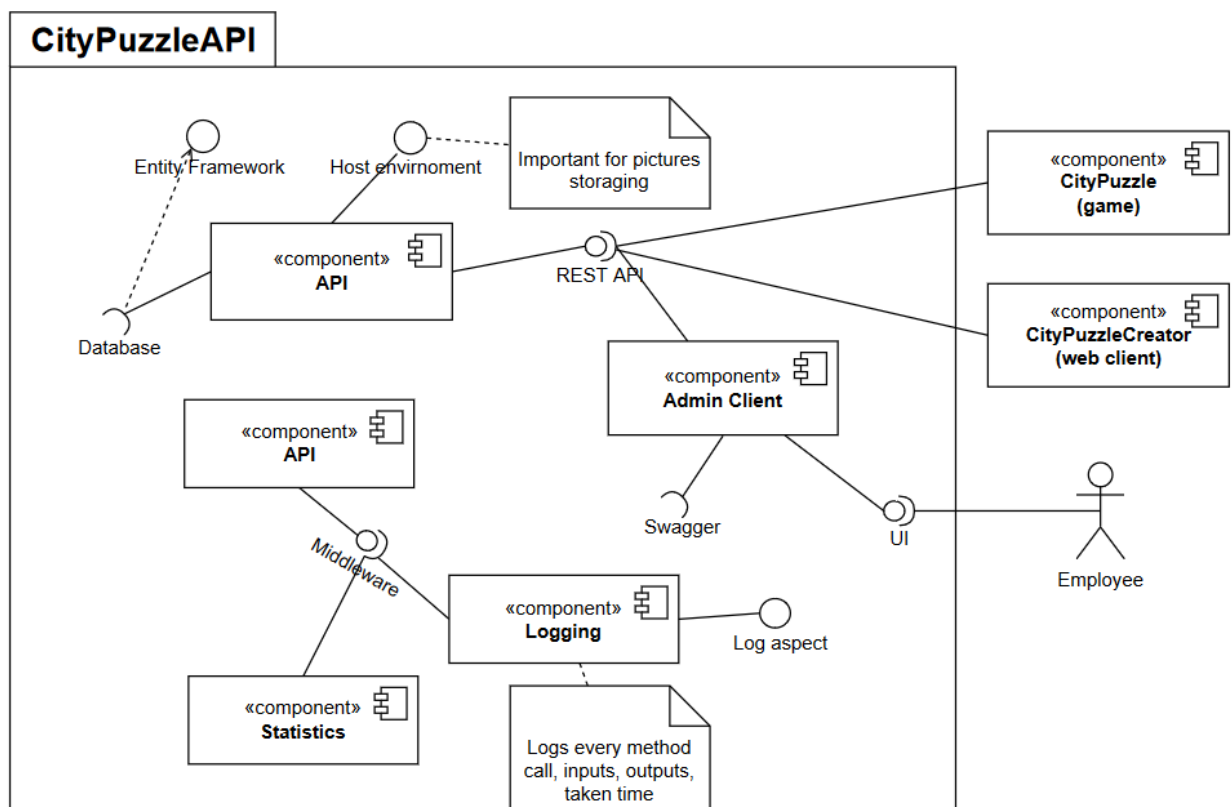


2.2. Development view

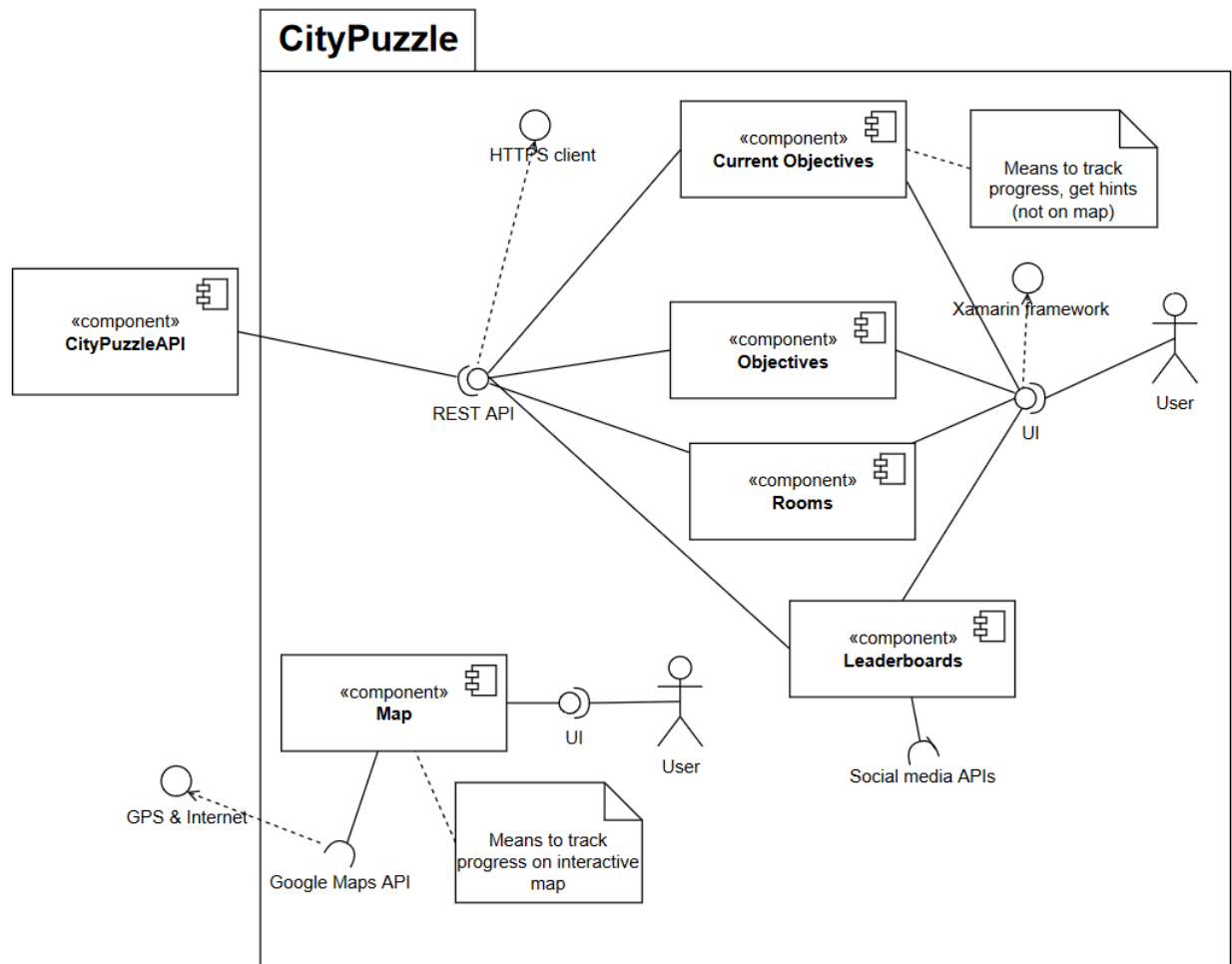
Pagrindiniai pokyčiai „development“ požiočiu buvo Web kliento sukūrimas ir jo pritaikymas esamam API. Taip pat reikėjo ir esamą API patobulinti naujai architektūrai, kuri užduočių kūrėjų rolę iš Vadovų perdavė Klientams, Partneriams.



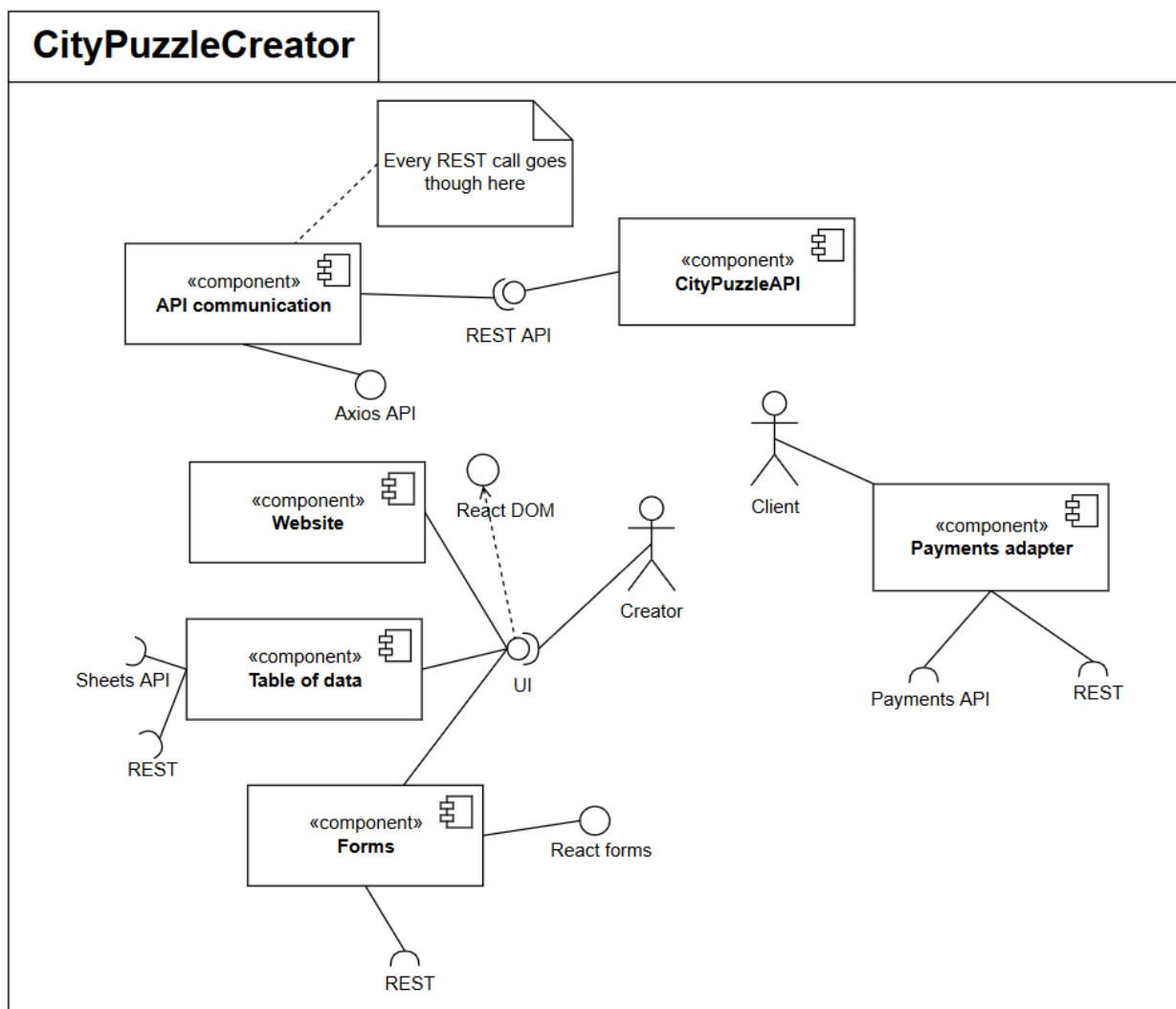
Apibendrinta sistemos architektūra susidaro iš 4 pagrindinių komponentų – Duombazės, API, žaidimo ir web kliento. Komponentai, su kuriais dirba Partneriai/Klientai/Vartotojai, susisiečia su Duombaze per API tarpininką. Ši architektūra leidžia komandai užtikrinti efektyvesnę veikimą („back ups“, populiariausių vietų saugojimas) ir saugumą (tiesiogiai nepasiekiamo duombazė). Senesnė architektūra buvo panaši, tik trūko „web“ kliento.



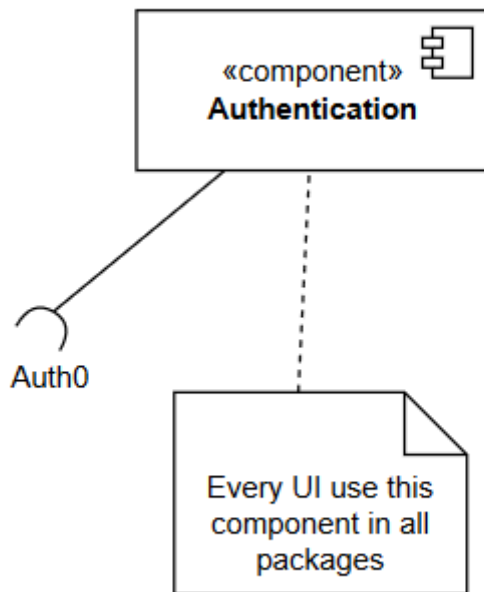
API yra tarpininkas tarp duombazės ir žaidimo/puslapio. Su Duombaze bendraujama su Entity Framework pagalba (tai lengviau negu daryti grynas SQL uzklausas), joje saugoma tekstinė informacija. Nuotraukų saugojimui pasitelkiame ASP.NET „host environment“. Su kita puse bendraujame REST API pagalba, kuri kiti komponentai turi tureti. Duombazes valdymui pasitelkiame admin klientą, kuris turi implementuoti „Swagger“ iranki (jis palengvina darbą su DB). Kiti komponentai irgi gauna/siunčia informaciją su REST. Taip pat turime Middleware interfaciā, kurio reikia programos sekimui. Tai naudinga stebėti API efektyvumą ir galimai įsilaužimus.



Žaidimas iš „development“ perspektyvos, dėl REST, yra priklausomas nuo HTTP kliento, kuris ir yra šiame pakete. Komponentai kaip „Current Objectives“, „Objectives“, „Rooms“, „Leaderboards“ turi UI interfeicą, bet turi implementuoti REST, dėl susisiekimo su API ir DB. „Current objectives“ padengia užduotis, kurias vartotojas jau daro, „Objectives“ – užduotys nedaromos, „Rooms“ – komandai sudaryti, „leaderboards“ susisies su socialiniais tinklais, nes iš mąstydami verslo kontekstą pirmame laboratoriniame, supratome kad vartotojai nori rungtyniauti. Taip pat turime „Map“, kuris turi implementuoti Google Maps API. Šis yra priklausomas nuo GPS ir Interneto, kurie jau nėra programėlės dalis.



Šis paketas atsakingas už klientų ir partnerių užduočių kūrimo ir sekimo galimybes. API komunikacijos komp. turi Axios API, kuris palengvina darba su HTTP/REST ir komunikacija su API. Klientai turi susimokėti už vietą, todėl jiems priskiriamas mokejimo komponentas ir jis praneš API apie to įvykdymą. Partneriai neturi susimokėti. Kūrėjai gali per UI pasiekti puslapį, lentelę su pasirinkimais keisti duomenis ir formą, kuri leis juos idėti, Reac Forms pagalba.

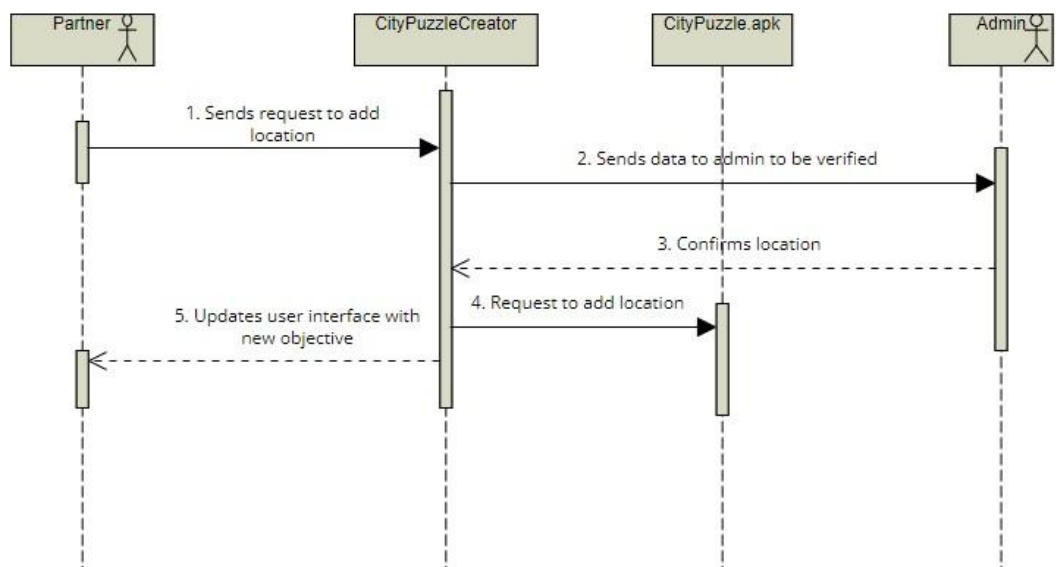


Galutinis komponentas – autentikacija. Ji svarbi saugumo atžvilgiu daugumai „Stakeholders“. Dėl to ji naudojama visur, kuri tik gali prieiti vartotojas/klientas/darbuotojas/partneris/kita.

2.3. Process view

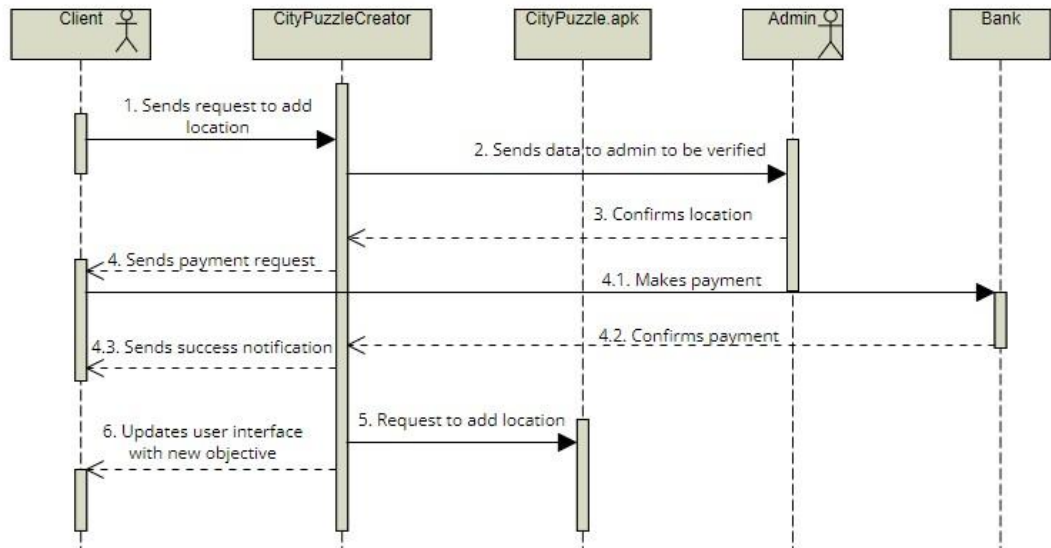
“Process view” lygis detaliai parodo kaip sistemos procesai veikia. Šiame dokumente vaizduojame kelis procesus: naujo objekto pridėjimo tiek partneriui, tiek klientui, pridėtos nuotraukos validavimo, prisijungimo klaidos, bendro objekto pridėjimo procesą.

2.3.1.



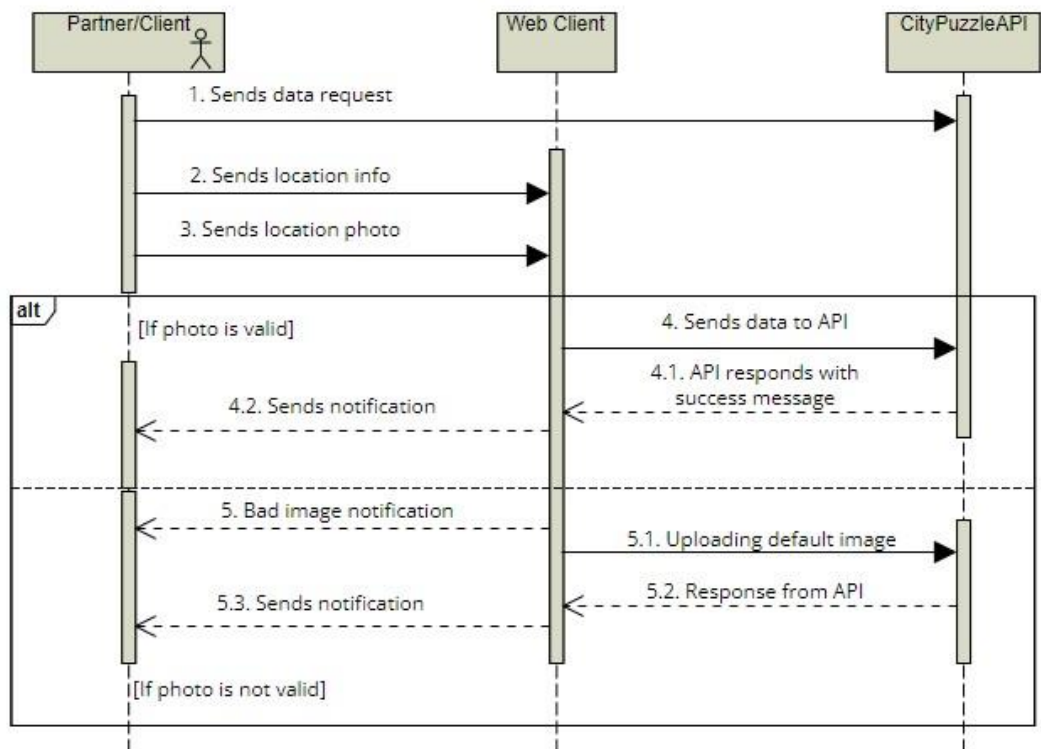
2.3.1. Diagrama pavaizduoja objekto pridėjimo procesą partneriui. Partneriai (t.y. valstybinės organizacijos ir t.t.) gali teikti prašymą objekto pridėjimui per CityPuzzleCreator web clientą, tačiau visos vietos, turi būti patikrintos darbuotojo.

2.3.2.



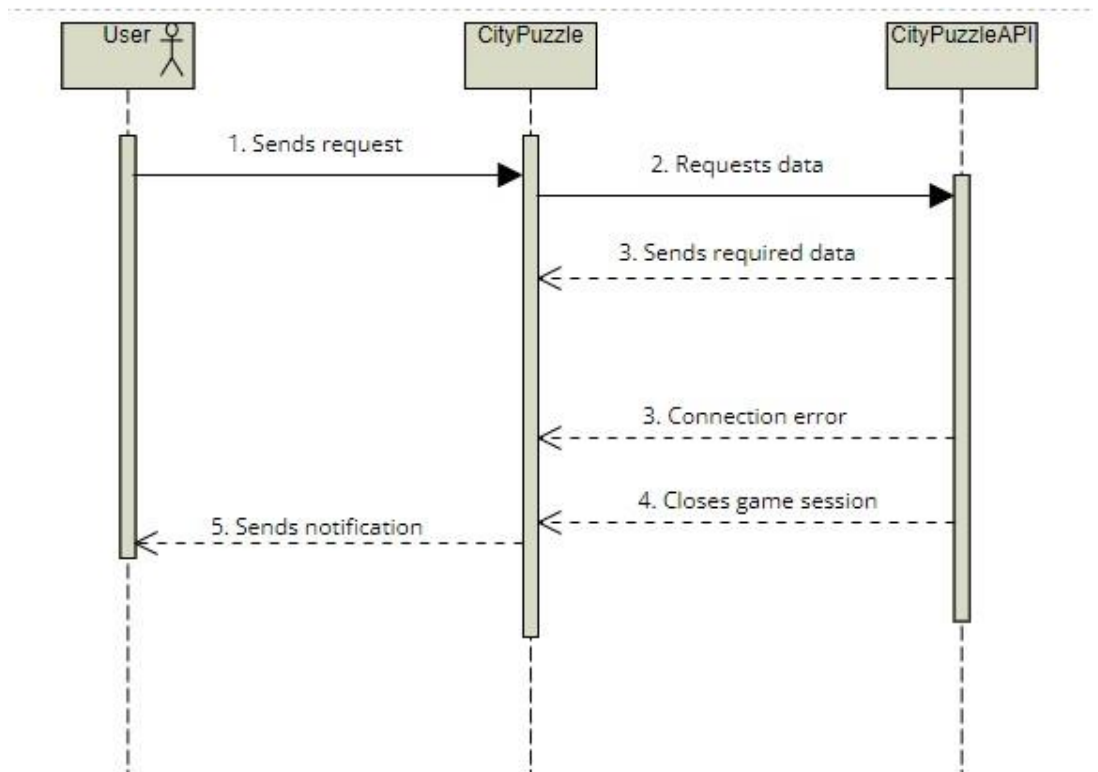
2.3.2. Diagrama pavaizduoja panašų objekto pridėjimo procesą kaip ir 2.3.1 diagramoje, tačiau jau klientams. Esminis skirtumas, tarp šių dviejų diagramų, yra mokėjimo proceso pridėjimas. Visos bandomos pridėti vietos taip pat yra patikrinamos.

2.3.3.



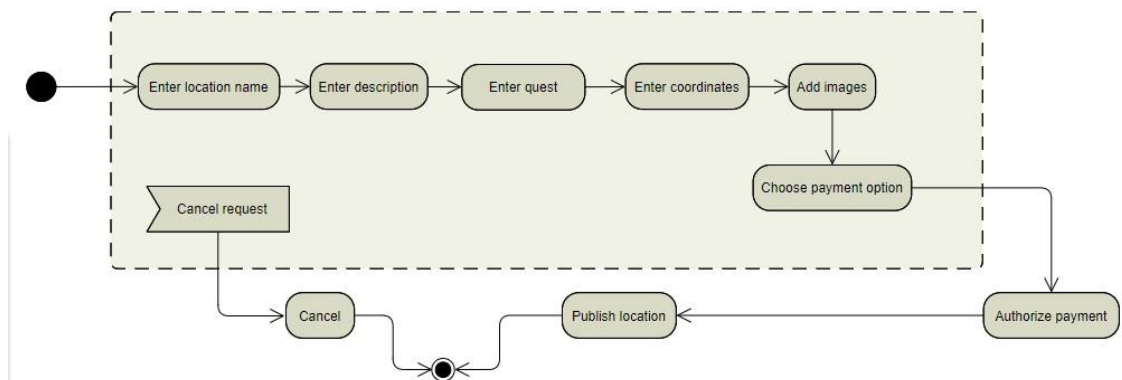
2.3.3. Diagrama vaizduoja kliento arba partnerio įkeltos fotografijos validavimo procesą. Jeigu, įkelta vietos nuotrauka, viršija ar neatitinka nustatytų standartų, t.y., nuotrauka turėtų būti 4:3 santykio ir neturėtų viršyti 3MB dydžio. Atveju, jei bandoma įkelti nuotrauka yra didesnio dydžio ar neatitinka 4:3 santykio, klientas/partneris yra informuojami, ir yra automatiškai įkeliami laikina default nuotrauka.

2.3.4.



2.3.4. Diagramoje yra vaizduojamas atvejis, kai atsitinka prisijungimo klaida (pvz. dingsta ryšys). Tokiu atveju, yra pranešama naudotojui ir yra automatiškai uždaroma žaidimo sesija.

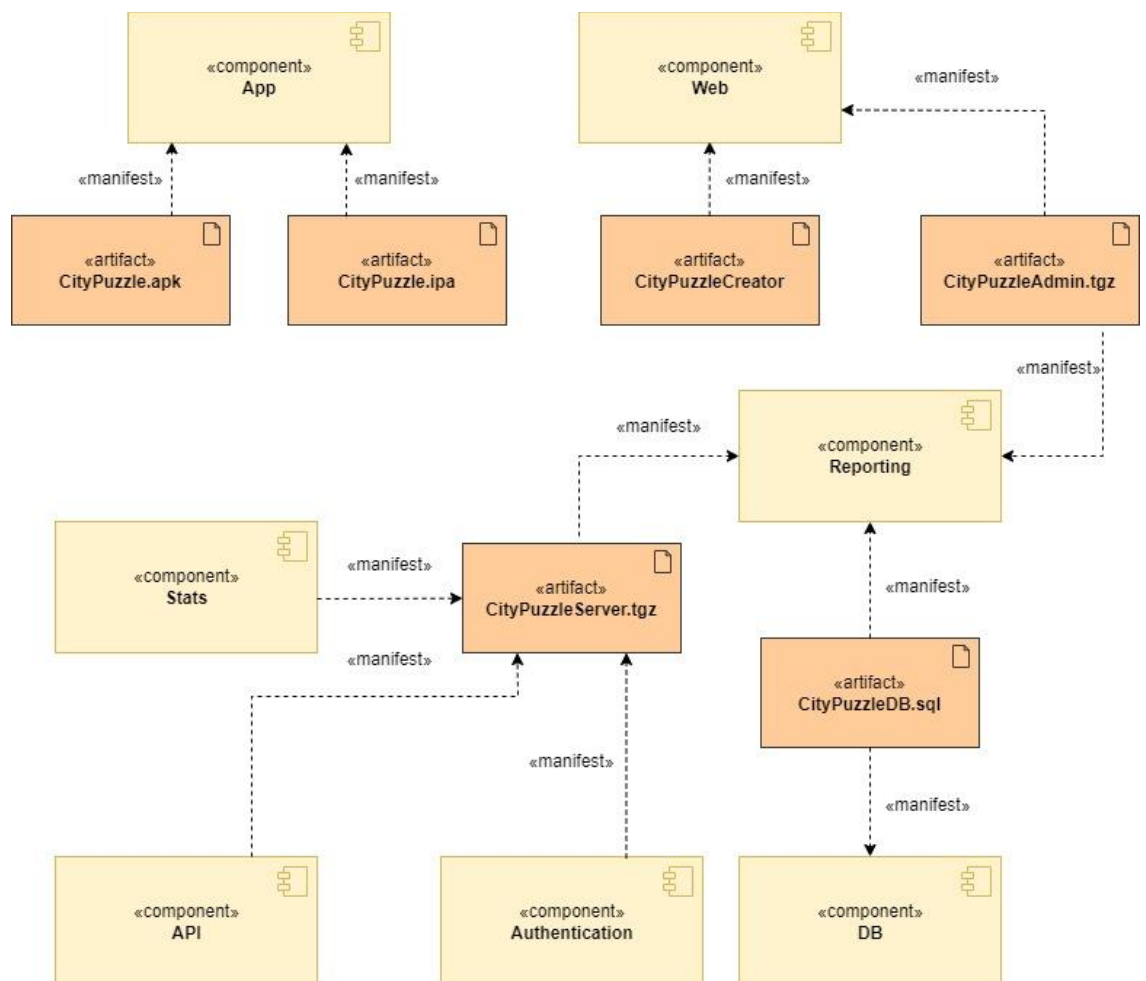
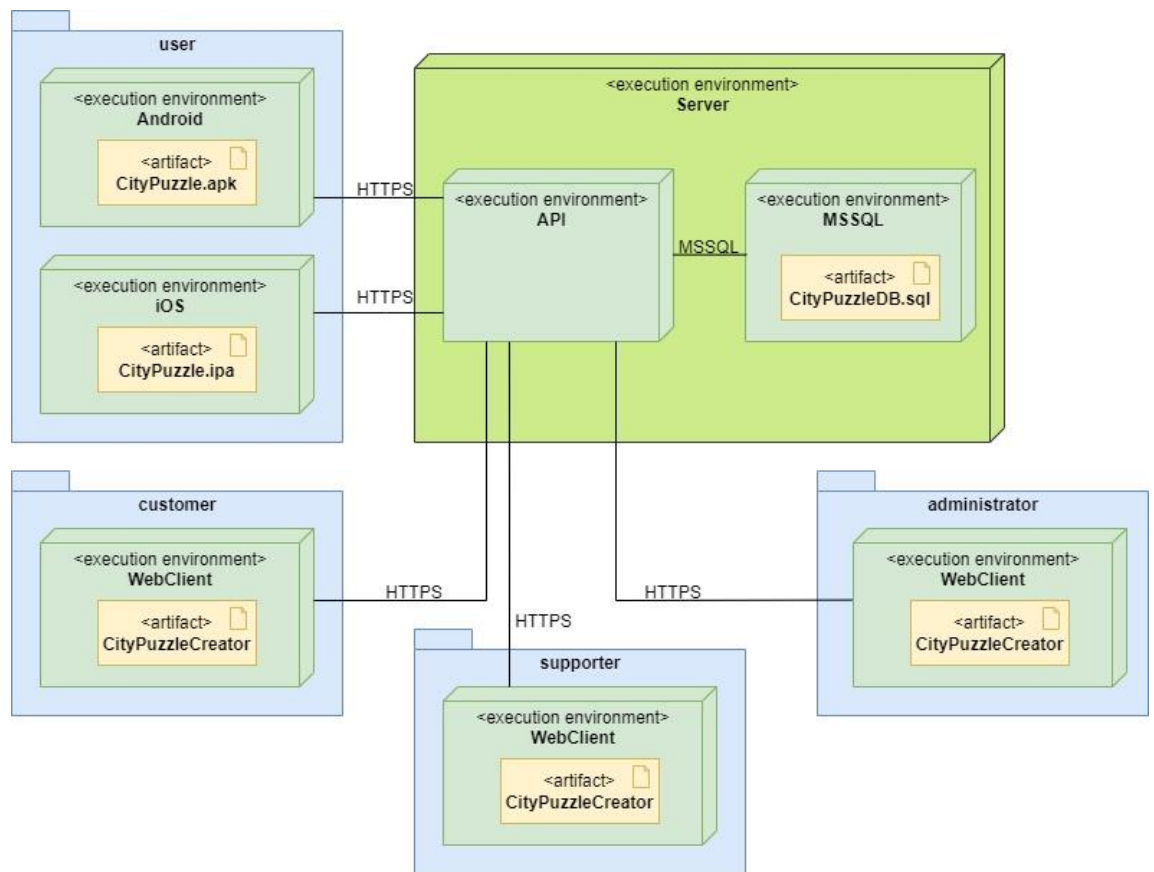
2.3.5.



2.3.5. Diagramoje yra pavaizduotas bendras objekto pridėjimo procesas klientams. Vertėtų pažymėti, jog bet kada galima paspausti „cancel“ mygtuką, kuris nutraukia procesą.

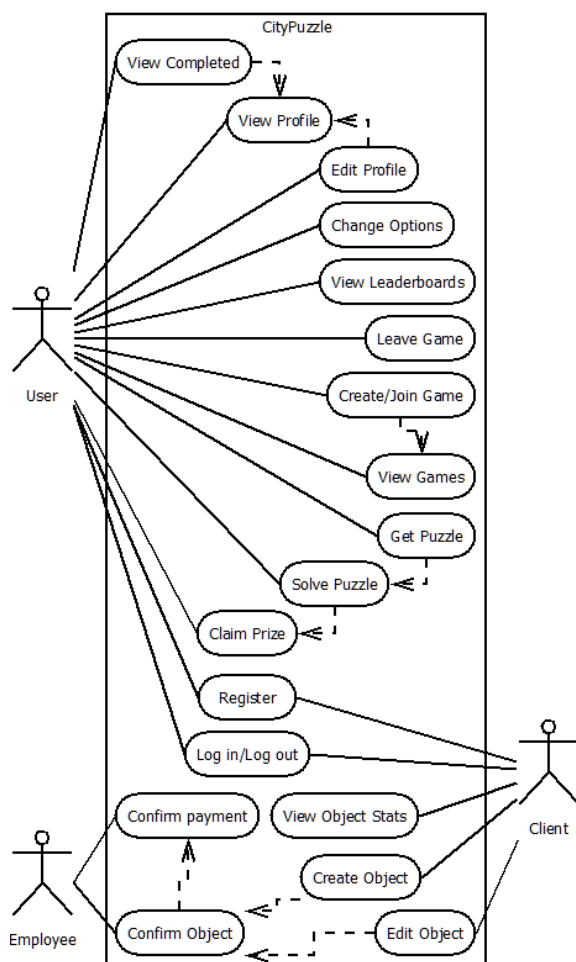
2.4. Physical view

„CityPuzzle“ žaidimas vartotojams (angl. user) pasiekiamas Android ir iOS operacinėse sistemose, tuo tarpu klientas (angl. customer), rėmėjas (angl. supporter) ir sistemos administratorius (angl. administrator) prie „CityPuzzleCreator“ aplinkos, patalpintos internetinėje svetainėje, jungiasi per bet kokią naršyklę. API ir MSSQL duomenų bazė komunikuoja serveryje.



2.5. Use Case View

2.5.1. Diagramoje nurodyti visų sistemos vartotojų galimi veiksmai kartu su procesais, nuo kurių jie priklauso.



Atsekamumas

naudodami atsekamumo matricą patikrinom ar sistemos komponentai “development view” dalyje logiškai siejasi su sistemos artefaktais.

Naudodami atsekamumo matricą patikrinom, ar sistemos komponentai “physical view” dalyje logiškai siejasi su sistemos artefaktais.

	App	Web	Stats	Reporting	API	Authentication	DB
CityPuzzle.apk	X						
CityPuzzle.ipa	X						
CityPuzzleCreator		X					
CityPuzzleAdmin.tgz		X		X			
CityPuzzleServer.tgz			X	X	X	X	
CityPuzzleDB.sql				X			X

Github:

CityPuzzleAPI: <https://github.com/JustasBan/CityPuzzleAPI>

CityPuzzle: https://github.com/JustasBan/Skestam_Kartu

CityPuzzleCreator: <https://github.com/JustasBan/CityPuzzleCreator>