Факултет техничких наука

Универзитет у Новом Саду

Виртуелизација процеса

Прорачуни на основу захтева
Техничка документација

Аутори:
Сања Јовишић
Данијел Јовановић
Катарина Маринковић
Андреа Чукић

Број индк:
PR25/2020
PR55/2020
PR134/2020
PR145/2020

25. мај 2023.

Садржај

1 Опште информације о пројекту

Прорачуни на основу захтева је пројекат из предмета Виртуелизација Процеса који се слуша у VI семестру на смеру Примењено софтверско инжењерство на Факултету техничких наука у Новом Саду.

Апликација прикупља податке од корисника о остварeним вредностима потрошње електричне енергије, по сатима, за данашњи дан.

Апликација се састоји од сервисне и клијентске апликације као и дистрибуираног сервиса XML базе података. Сервисна и клијентска апликације комуницирају путем WCF-а. Клијентска апликација је конзолна апликација која има опцију уноса наредбе која се прослеђује сервису.

Апликација функционише тако што, на самом почетку, корисник сачува CSV датотеку о остварeним вредностима потрошње електричне енергије, по сатима, за данашњи дан, на дефинисаној локацији на рачунару на ком се налази клијентска апликација. Затим се датотека шаље сервисној апликацији, и избрисана је са локације где је претходно сачувана.

Сервисна апликација чува пристигле податке у бази података, на локацији која је дефинисана у конфигурационој датотеци сервисне апликације. Ако се за неки сат уочи грешка, креира се нови ***Audit*** објекат. Audit објекти се уписују у базу података и прослеђују се клијентској апликацији у оквиру резултата. Текстови грешака из Audit објеката исписују се на конзоли клијентске апликације. Грешка се може десити ако је податак невалидан, ако недостају неки подаци или ако постоји грешка у структури CSV датотеке.

Такође, корисник може од сервисне апликације затражити и прорачун за минималну и максималну потрошњу или стандардну девијацију. Када су извршени поменути прорачуни, кориснику је прослеђена текстуална датотека (**.txt**).

2 Техничка и контекстуална спецификација пројекта

2.1 Коришћење технологије у изради пројекта

- C# 7 у оквиру .NET Framework развојног окружења верзије 4.8
- XML датотеке за перзистенцију података
- Windows Communication Foundation

2.2 Контекст проблема

- Свакодневница нам доноси разне проблеме везане за бележење већег скупа података. Број корисника свакодневно расте, док се неки рутински послови могу полу-аутоматизовати.

  Тако прикупљени подаци нису од великог значаја јер фирма за дистрибуцију електричне енергије не може направити статистички преглед и своју будућу дистрибуцију подесити правилно, као и оптимизовати на начин да сви корисници и даље остану задовољни.

2.3 Информационо-апликациони опис

- Целокупно идејно решење састоји се од више компоненти односно **микросервиса** који могу радити заједно у склопу, али их одликује и потпуна независност од остатка информационог система. Предуслов је да се покрену сервисна и клијентска апликација као и сервер базе података.

Common Class Library

Компонента која је заједничка за све компоненте система. Садржи датотеке, изузетке и моделе. Датотека је класа која моделује рад са датотекама по ***Disposable*** шаблону ослобађања меморије.

Класа изузетака се састоји од специјално прилагођених изузетака, посебне намене за прикупљање одређених изузетака који се могу десити, а нису подржани у оквиру .NET системског окружења.

Класа модела садржи прецизно моделоване класе за податке који ће се користити за потребе уписа грешке, чување ентитета из базе података, као и прорачуна позива делегата.

Захваљујући наведеној компоненти самој апликацији је омогућено да успостави конекцију са свим осталим компонентама.

Интерфејси

1. IRadSaDatotekom (наслеђује IDisposable)
Интерфејс моделује рад са датотеком по Disposable шаблону.

2. IPreglediPotrosnje
Интерфејс моделује методу која треба да на основу улазне колекције података за текући дан, пронађе минимум и/или максимум и/или потрошњу по стандардној девијацији.

3. IPreglediPotrosnje
Метода која као улазне параметре прима које вредности прорачуна клијент жели да изврши за текући дан.

Повратна вредност је текстуална датотека која је креирана од стране сервиса и садржи преформатиран испис прорачуна. У случају да за тренутни дан нису забележени подаци потрошње или потрошњу није могуће израчунати дешава се изузетак ***PregledPotrosnjeIzuzetak***.

Klijent
Компонента која моделује сервис интеракција и интеграција. Посредством дате компоненте омогућено је да корисник сачува CSV датотеку (чим се пошаље датотека се брише са рачунара са којег је послата), која садржи податке потрошње електричне енерије по сатима, за данашњи дан.

Омогућава да корисник добије информације о минималној и максималној потрошњи текућег дана, као и о стандардној девијацији потрошње. Компонента није чвориште података, већ само привремено чува податке, док се датотека не пошаље или по позиву наредбе користећи неке од три дата параметра.

Интерфејси

1. IKomanda
Интерфејс моделује методе за **Get** и **Send** команде.

bool SlanjeCsv(out List<Audit> greske)
Метода на основу локација директоријума CSV датотека из ***App.config*** креира нови ***MemoryStream*** у ком је отворена датотека и тај ток података прослеђује сервису за парсирање и обраду датотеке.

Одредишна тачка приступа је метода позвана проксијем за парсирање датотеке.

Повратна вредност методе парсирање је ***true*** ако је успешно креиран MemoryStream на основу датотеке из предефинисаног директоријума и послат на сервис, у супротном ***false***.

У случају да се десио ***timeout*** (сервис није покренут/доступан) изазвати изузетак ***KomandaIzuzetak***.

bool SlanjeGet(bool IsMin = false, bool IsMax = false, bool IsStand = false)
Метода која шаље **Get** команду на сервис, улазни параметри функције су подразумевано ***false***, док се могу истовремено захтевати и све три калкулације.

Одредишна тачка приступа је метода прорачуна лоцирана на статистичком сервису.

Повратна вредност је ***true*** ако постоји барем један запис, у супротном ***false***. Ако су сва три параметра ***false*** (корисник је унео само Get, изазвати изузетак ***KomandaIzuzetak***.
На крају ако постоји податак, креира се текстуална датотека која садржи примљене прорачуне.

2. IMeni

Интерфејс моделује мени који корисник користи како би остварио интеракцију са сервисом статистике и обраде података.

void IspisiMeni()
Метода која моделује испис менија који кориснику нуди опције које може да користи.

void MeniSend()
Метода која моделује опције слања података на сервер кроз мени.

void MeniGet(string unos)
Метода која моделује прихватање опција из менија које је корисник унео.

3. IUpisUIzvestaj

Метода која као улазне параметре прима које вредности прорачуна клијент жели да изврши за текући дан.

Повратна вредност је текстуална датотека која је креирана од стране сервиса и садржи преформатиран испис прорачуна. У случају да за тренутни дан нису забележени подаци потрошње или потрошњу није могуће израчунати дешава се изузетак PregledPotrosnjeIzuzetak.

Server

Компонента која врши прикупњање података и омогућава преглед потрошње електричне енергије. Такође извршава прорачуне минималне и максималне потрошње као и прорачун стандардне девијације за текући дан и клијенту враћа форматирану текстуалну датотеку са захтеваним подацима.

Интерфејси

1. IDataFetcher

Интерфејс моделује методу потребну за прикупљање података.

IEnumerable<Load> PrikupiPodatkeZaTekuciDan()
Метода се повезује са сервером XML базе података и прикупља све податке за текући дан и смешта у колекцију података.
Повратна вредност методе је листа свих података мерења.

2. IDevijacija

Интерфејс који моделује методу за прорачун девијације вредности.

double StandardnaDevijacija(IEnumerable<double> merenja)
Метода која рачуна стандардну девијацију по формули.
Улазни параметар је листа мерења, док је излаз израчуната стандардна девијација.

3. IIzvestajProracuna

Интерфејс моделује генерисање извештаја прорачуна након инвокације делегата/догађаја.

IRadSaDatotekom NapraviIzvestajProracuna(IEnumerable<Proracun> podaci)
Метода која се позива након позива делегата метода прорачуна.

Повратна вредност је ***ток бајтова*** на тексутални фајл у којем су уписани прослеђени прорачуни у формату: ***Tip_Proracuna: Vrednost_Proracuna***.
Док су улазни параметри листа прорачуна која се уписује у излазни txt фајл.
У случају да је листа прорачуна празна или неки од података невалидан дешава се ***IzvestajIzuzetak***.

XML Baza Podataka

Компонента која ради са XML базом података. Омогућава да се у базу података ажурирају подаци прослеђени из CSV датотеке, да се CSV датотека парсира, као и да се грешке приликом парсирања упишу у базу података. Такође, омогућено је читање података из базе података за текући дан.

Интерфејси

1. IBazaPodataka

Интерфејс који моделује рад са XML базом података.

int UpisUBazuPodataka(List<Load> podaci, List<Audit> greske)
Повратна вредност је број успешно уписаних редова у базу података ***TBL_LOAD.xml***, ако ништа није уписано враћа 0. У базу података TBL_LOAD.xml уписују се ***Load***, док се у табелу ***TBL_AUDIT.xml*** уписују Audit подаци из листи.

void ProcitajIzBazePodataka(out List<Load> procitano)
Ако је база података празна или није прочитан ниједан податак враћа ***празну листу***, у супротном ***ишчитану листу података***.
Из базе података ***TBL_LOAD.xml*** ишчитава се ред по ред података и креира објекат ***Load*** и чува у листу прочитаних.
У случају да датотека ***TBL_LOAD.xml*** не постоји креира се нова ***празна*** XML датотека.

2. IXmlCsvFunkcije

Интерфејс моделује парсирање CSV датотеке и отварање датотеке.

bool ParsiranjeCsvDatoteke(MemoryStream csv, out List<Audit> greske)
Повратна вредност је ***true*** ако се није десила ниједна грешка приликом обраде csv датотеке, у супротном ***false***.
Парсира се улазна датотека, уколико датотека не постоји, унет је погрешан тип податка, погрешан датум креира ***Audit*** објекат и додаје га у листу

грешака (касније се прослеђује клијенту) ако је парсирани ред валидан, креира се нови објекат класе Load и смешта у привремену листу Load објеката. Метода позива посебну методу за упис у базу података.

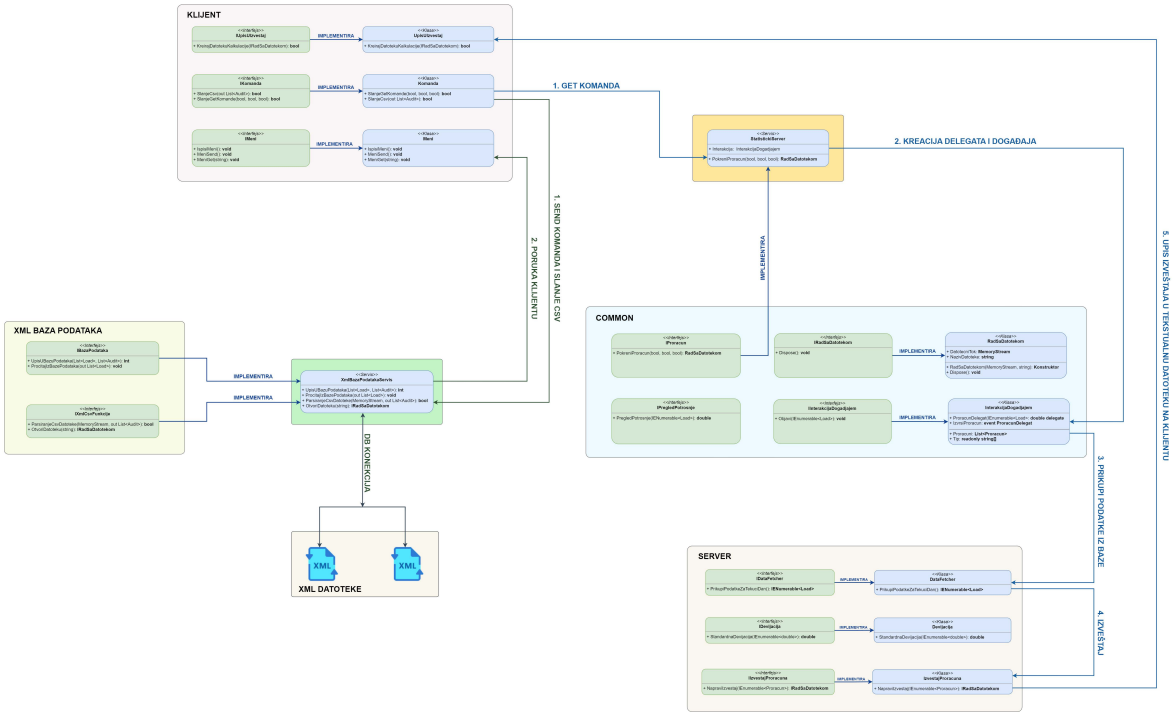
IRadSaDatotekom OtvoriDatoteku(string putanja_datoteke)
Ако xml датотека постоји и успешно се отвори враћа ***true***, у супротном false. Путања датотеке је предефинисана у ***App.config***.
Ако датотека не постоји креира се нова ***празна*** XML датотека.

3 Дизајн и архиктетура система

3.1 Уопштење

- Дизајн система је процес дефинисања елемената система као што су модули, архитектура, компоненте и њихови интерфејси и подаци за систем на основу специфицираних захтева.

3.2 Компоненте и интерфејси система



4 Сумативна вредност и проактива пројекта

Пројекат има за циљ да на потпуно нови начин моделује изрази промену начина на који се бележи потрошња за текући дан.

Фаза 1: Развој апликације – унапређење корисничког искуства

У наредном периоду, фокус ће бити на развоју и унапређењу серверске апликације. Фокус ће бити на наставак истраживања и развоја нових функционалности и могућности које ће омогућити крајњим корисницима да прецизно бележе своју потрошњу за текући дан. Клијентска апликација ће добити потпуно нови интерфејс, како би сам екосистем постао неопходан алат за ефикасно управљање потрошњом.

Фаза 2: Интеграција са паметним уређајима

Интеграција апликације са паметним уређајима представља још један од планова како би се пружила још већа лакоћа коришћења саме апликације. Овакав вид интеграције ће омогућити корисницима да аутоматски прате своју потрошњу путем својих уређаја и да добију ***realtime*** информације о својим потрошњи.

Фаза 3: Употреба у SCADA системима - аутоматизација прикупљања података

Како су SCADA системи скалабилни, уз одговарајуће подешавање, и једноставни за употребу када је индустријске процесе потребно аутоматизовани и надгледати. Апликацију односно део апликације који прикупља податке и захтева интеракцију корисника био би пребачен на SCADA подсистем који би мерења прикупљао и аутоматски слао на обраду сервису, те би сам терет са корисника био сведен на проактивно надгледање параметара система, и захтевање прорачуна.

Фаза 4: Анализа података и персонализоване препоруке

Кроз употребу напредних алгоритама и технике машинског учења, намера је да се у оквиру апликације и целог екосистимеа развију аналитички механизми како би се додала могућност да се апликација прилагођава кориснику на начин који је он користи и коју му олакшава свакодневну употребу. Механизми ће анализирати бележене податке о потрошњи, начине и шаблоне употребе апликације, као и најчешће захтеване прорачуне и радити активне позадинске прорачуне и исте кеширати како би на сам захтев, крајњи прорачун био доступнији у краћем временском периоду.

Фаза 5: Дистрибуираност система

Како би се осигурала већа поузданост и доступност система за бележење потрошње за текући дан, план је имплементирати додатну дистрибуирану архитектуру као и имплентирати механизме репликације. Крајњи исход оваквог вида приступа ће резултирати смањеним ризиком од кварова и повећаном отпорношћу на оптерећење.

Дистрибуирањем система на више локација или прелазак на ***cloud*** платформу, обезбедиће корисницима приступ услугама независно од географске локације, уз минимално време захтев-одговр и максималну поузданост.

Предност која се добија је и ***скалабилност система*** - већим бројем корисника и растућим обимом података систем ће бити у стању да се лако прилагоди ново насталој. Дистрибуирани систем ће бити дизајниран и имплементиран уз коришћење напредних технологија и сигурносних механизама како би се заштитила приватност и интегритет корисничких података.

Закључак

Може се закључити да је у самом старту пројекат успешно постигао своје иницијалне циљеве и достигао значајан напредак. Развојни пут пројекта представљао је изазовно путовање које је захтевало тимски рад, стратешко планирање и упорност.

Током овог пројекта, успели смо да идентификујемо кључне проблеме и изазове са којима смо се суочавали, као и да развијемо иновативно решење. Истраживање је било темељито и свеобухватно, те нам је омогућило да стекнемо дубље разумевање проблема и правовремено реагујемо на промене.Постигли смо важне кораке у иновацији и усвајању нових технологија.

Прорачуни на основу захтева је био платформа за истраживање и имплементацију нових идеја, унапређење постојећих решења и корак напред као основа за будућа решења у индустрији.