

FOYDALANUVCHIDAN MA'LUMOT OLISH VA UNI QAYTA ISHLASH DASTURI YOZISH

Mirsaid Yusupov Abdulaziz o'g'li

*Farg'ona davlat universiteti Amaliy matematika
va informatika kafedrası o'qituvchisi*

E-mail: mirsaidbeky@gmail.com

Abduraxmonova Nozanin Robiljon qizi

*Farg'ona davlat universiteti Amaliy matematika
yo'nalishi 3-bosqich 23.07-guruh talabasi*

E-mail: abduraxmonovanozanin91@gmail.com

Annotatsiya: Mazkur ilmiy ish zamonaviy dasturiy ta'minot tizimlarining fundamental komponentlaridan biri bo'lgan foydalanuvchi ma'lumotlarini olish, verifikatsiya qilish va samarali qayta ishlash jarayonlarini kompleks tahlil qilishga qaratilgan. Maqolada ma'lumotlarni kirish interfeyslari orqali to'plashning me'moriy yechimlari, xususan, sinxron va asinxron yondashuvlarning o'ziga xos jihatlari batafsil ko'rib chiqiladi. Shuningdek, ma'lumotlarning to'g'riligi va xavfsizligini ta'minlash uchun verifikatsiya va sanitariya (ma'lumotlarni tozalash) mexanizmlarining ilmiy asoslari tahlil qilinadi. Qayta ishlash bosqichida qo'llaniladigan modulli yondashuv, xususan, mikro servislar kabi ilg'or arxitekturalarning foydalanuvchi ma'lumotlari oqimini boshqarishdagi ahamiyati chuqur o'rganiladi. Ushbu tadqiqot dasturiy ta'minot muhandislari, o'qituvchilar va magistrantlar uchun ma'lumotlarni boshqarish tizimlarini loyihalashda tizimli bilim manbai bo'lib xizmat qiladi.

Kalit So'zlar: Ma'lumotlarni Qayta Ishlash, Foydalanuvchi Interfeysi, Dasturiy Me'morchilik, Ma'lumotlar Oqimi, Verifikatsiya, Sanitariya, Modulli Dizayn.

Аннотация: Настоящая научная работа посвящена комплексному анализу процессов сбора, верификации и эффективной обработки пользовательских данных, являющихся фундаментальным компонентом современных программных систем. В статье подробно рассматриваются архитектурные решения для сбора данных через интерфейсы ввода, включая специфику синхронного и асинхронного подходов. Также анализируются научные основы механизмов верификации и санитарии (очистки данных), необходимых для обеспечения достоверности и безопасности информации. В разделе обработки данных углубленно изучается значение модульного подхода и передовых архитектур, таких как микросервисы, в управлении потоком пользовательских данных. Данное исследование послужит систематическим источником знаний для инженеров-программистов, преподавателей и магистрантов при проектировании систем управления данными.

Ключевые Слова: Обработка Данных, Пользовательский Интерфейс, Программная Архитектура, Поток Данных, Верификация, Санитария, Модульный Дизайн.

Abstract: This scientific paper is dedicated to the comprehensive analysis of processes involved in collecting, validating, and efficiently processing user data, which represents a fundamental component of modern software systems. The article thoroughly examines architectural solutions for data acquisition via input interfaces, detailing the specifics of synchronous and asynchronous approaches. Furthermore, it analyzes the scientific basis of verification and sanitization mechanisms necessary to ensure data accuracy and security. The section on data processing delves into the significance of modular design and advanced architectures, such as microservices, in managing the flow of user data. This research is intended to serve as a systematic knowledge source for software engineers, educators, and graduate students when designing data management systems.

Key Words: Data Processing, User Interface, Software Architecture, Data Flow, Verification, Sanitization, Modular Design.

Kirish

Zamonaviy axborot texnologiyalari va dasturiy ta'minot tizimlarining kundalik faoliyatida foydalanuvchidan ma'lumot olish va ularni qayta ishlash jarayoni fundamental va hal qiluvchi ahamiyatga ega. Bugungi kunda yirik miqyosdagi ilovalar va xizmatlar insonlarning o'zaro aloqasidan, sensorlardan va raqamli interfeyslardan oqib kelayotgan ulkan ma'lumotlar oqimini doimiy ravishda boshqarishni talab qiladi. Ma'lumotlar nafaqat tizimning ishlashi uchun zarur bo'lgan resurs, balki xizmatlar sifatini oshirish, foydalanuvchi xatti-harakatlarini tahlil qilish va biznes qarorlarini qabul qilish uchun strategik ahamiyatga ega aktiv hisoblanadi.

Biroq, bu jarayon o'ziga xos me'moriy qiyinchiliklarga ega. Birinchidan, ma'lumotlarning hajmi (Volume) doimiy ravishda o'sib bormoqda. Ikkinchidan, ma'lumotlarning tezligi (Velocity) real vaqt rejimida qayta ishlashni talab qiladi. Uchinchidan, ma'lumotlarning xilma-xilligi (Variety), ya'ni matn, raqam, media yoki sensor ma'lumotlari kabi turli formatlardagi ma'lumotlarni qabul qilish va birlashtirish zarurati murakkablikni oshiradi. Natijada, dasturiy ta'minot tizimi ma'lumotlarni samarali olish, ularning ishonchliligi va xavfsizligini kafolatlash, shuningdek, qayta ishlash operatsiyalarini kengaytiriluvchan (scalable) tarzda amalga oshirish uchun mustahkam me'moriy yechimlarga ega bo'lishi shart.

Ushbu ilmiy tadqiqotning asosiy maqsadi foydalanuvchidan kiruvchi ma'lumotlarni qabul qilishdan tortib, ularni yakuniy saqlashgacha bo'lgan butun siklni qamrab oluvchi tizimli, ishonchli va kengaytiriluvchan dasturiy yechimlarni loyihalashning ilmiy-nazariy asoslarini tahlil qilishdan iborat. Biz ma'lumotlarni

olishda qo'llaniladigan sinxron va asinxron yondashuvlarning me'moriy ta'sirini, kirish ma'lumotlarining xavfsizligi va to'g'riligini ta'minlashdagi verifikatsiya va sanitariya mexanizmlarining ahamiyatini chuqur o'rganamiz. Shu bilan birga, qayta ishlash bosqichida modulli dizayn va mikroservislar kabi ilg'or me'morchilik tamoyillarining ishonchli ma'lumotlar oqimini yaratishdagi rolini yoritib beramiz.

Maqolaning tuzilishi o'quvchiga mavzuni mantiqiy ketma-ketlikda tushunish imkonini beradi. Dastlab, biz ma'lumotlarni olishning me'moriy asoslarini va interfeyslar orqali ishlash usullarini ko'rib chiqamiz. So'ngra, ma'lumotlar sifatini ta'minlashning himoya qatlamlari, ya'ni verifikatsiya va sanitariya masalalari yoritiladi. Keyingi bo'limlar qayta ishlash tizimining modulli dizayniga va ma'lumotlarni saqlash strategiyalariga bag'ishlanadi, bu esa loyihaning akademik va amaliy ahamiyatini oshiradi.

Ma'lumotlarni Olish

Foydalanuvchidan ma'lumot olish tizimining mustahkamligi va samaradorligi uning me'moriy yondashuvlariga bog'liqdir. Ma'lumotlarning dasturiy tizimga kirish nuqtasi, ularni qabul qilish usuli va protokoli tizimning umumiy ishlash tezligiga va barqarorligiga bevosita ta'sir etadi.

Ma'lumotlar dasturiy tizimga turli xil kirish interfeyslari orqali yetib kelishi mumkin. Har bir interfeys turi o'ziga xos me'moriy talablarni qo'yadi. Masalan, grafikli foydalanuvchi interfeyslari (GUI) orqali olingan ma'lumotlar inson tomonidan kiritilganligi sababli, shakl maydonlari darajasida birinchi darajali verifikatsiyani talab qiladi. Buyruq qatori interfeyslari (CLI) yordamida olingan ma'lumotlar esa ko'pincha skriptlar yoki avtomatizatsiyalangan jarayonlar orqali yuboriladi, bu esa formatning aniq rioya etilishiga yuqori e'tibor qaratishni talab qiladi. Eng muhimi, Dasturiy Interfeyslar (API) orqali olingan ma'lumotlar boshqa dasturiy tizimlar yoki xizmatlar bilan bevosita aloqani ta'minlaydi. API ma'lumotlar olishning eng keng tarqalgan professional usuli bo'lib, ular uchun kirish nuqtalarining (endpoints) xavfsizligi va belgilangan protokollarga muvofiqligi asosiy me'moriy vazifadir.

Ma'lumotlarni qabul qilishda dasturiy ta'minot tizimlari sinxron yoki asinxron yondashuvlardan birini tanlaydi. Sinxron olish modelida dasturiy modul so'rovni qabul qiladi va javobni kutish vaqtida boshqa operatsiyalarni bajarmaydi, ya'ni so'rov to'liq bajarilmaguncha bloklanadi. Bu yondashuv oddiy stsensariylar va tezkor, kichik ma'lumotlar uchun mos keladi. Biroq, trafik yuqori bo'lganda, sinxron bloklanish tizimning o'tkazuvchanlik qobiliyatini sezilarli darajada pasaytirib, resurslarning samarasiz ishlatilishiga olib kelishi mumkin.

Aksincha, asinxron olish modeli tizimga so'rovni qabul qilish, uni navbatga qo'yish va so'rovni qayta ishlash natijasini kutmasdan darhol boshqa so'rovlarga xizmat ko'rsatish imkonini beradi. Bu yondashuv tizimning bir vaqtning o'zida ko'plab so'rovlarga xizmat ko'rsatish qobiliyatini keskin oshiradi. Asinxron me'morchilikda

ko'pincha I/O operatsiyalari uchun maxsus funksiyalar yoki ish zanjirlari (threads) qo'llaniladi, bu esa asosiy ish zanjirining bloklanishini oldini oladi. Yirik miqyosdagi veb-ilovalar va real vaqt rejimida ishlaydigan tizimlar uchun asinxron yondashuv kengaytiriluvchanlikni kafolatlovchi me'moriy ustunlikdir.

Ma'lumotlarning interfeyslar orqali muvaffaqiyatli uzatilishi aniq belgilangan protokollarga bog'liq. API orqali ma'lumot olishda HTTP kabi standart protokollar va JSON yoki XML kabi ma'lumot almashish formatlari asosiy rol o'ynaydi. Tizim bu protokollarga mos keladigan kirishni qayta ishlash uchun mo'ljallangan bo'lishi shart. Protokolning aniq belgilanishi va unga rioya etilishi nafaqat ma'lumotlarning to'g'ri olinishini, balki tizimlar o'rtasidagi o'zaro ishlash (interoperability) qobiliyatini ham ta'minlaydi, bu esa zamonaviy taqsimlangan tizimlarning ajralmas qismidir.

Ma'lumotlarning Ishonchliligi va Xavfsizligi

Foydalanuvchidan olingan har qanday ma'lumotning ishonchliligi, butunligi va xavfsizligi dasturiy ta'minot tizimining eng muhim me'moriy majburiyatlaridandir. Bu majburiyat verifikatsiya va sanitariya deb nomlanuvchi ikki asosiy jarayon orqali amalga oshiriladi.

Verifikatsiya Jarayonining Nazariy Asoslari

Verifikatsiya (tasdiqlash) bu ma'lumotlarning dasturiy tizimning ichki qoidalariga, cheklovlariga va kutilayotgan formatiga mos kelishini tekshirish jarayonidir. Verifikatsiya ikki darajada amalga oshiriladi:

1. Sintaktik Verifikatsiya: Ma'lumotning tuzilishi to'g'ri ekanligini tekshirish. Misol uchun, elektron pochta manziling standart formatga mos kelishi, raqamli maydonlarda faqat sonlarning mavjudligi yoki kiritilgan matnning ruxsat etilgan maksimal uzunlikdan oshmasligi.
2. Semantik Verifikatsiya: Ma'lumotning ma'nosi va mazmuni tizimning biznes qoidalariga mos kelishini tekshirish. Masalan, tizimga kiritilgan tug'ilgan sananing kelajak sanasi bo'lmasligi yoki ma'lumotlar bazasida allaqachon mavjud bo'lgan noyob identifikatorning qayta kiritilmasligi.

Bu jarayonlar ma'lumotlar qayta ishlash quvurining (pipeline) boshida, ya'ni ma'lumotlar asosiy tizim logikasiga kirishdan avval bajarilishi shart. Verifikatsiyani amalga oshirishning algoritmik usullari ichida Regulyar Ifodalar (Regular Expressions) va Schema Validation (Ma'lumotlar sxemasi asosida tekshirish) keng qo'llaniladi.

Sanitariya (ma'lumotlarni tozalash) jarayoni verifikatsiyadan keyin amalga oshiriladi va ma'lumotlarning xavfsizligini ta'minlashga qaratilgan. Sanitariya, ayniqsa, ma'lumotlar bazalariga kiritish yoki boshqa foydalanuvchilarga ko'rsatishdan oldin, kirish ma'lumotlaridan zararli yoki noo'rin tarkibni olib tashlashni o'z ichiga oladi. Eng keng tarqalgan xavfsizlik tahdidlaridan biri bu Saytlararo Skripting (Cross-

Site Scripting - XSS) hujumlari bo'lib, bunda foydalanuvchi ma'lumot kiritish maydoniga zararli dastur kodini (masalan, JavaScript) kiritadi.

Sanitariya mexanizmlari bu kabi zararli kodlarni avtomatik ravishda filtrlaydi yoki neytrallashtiradi (masalan, HTML teglarini maxsus belgilarga almashtirish orqali). Ma'lumotlarni tozalash, shuningdek, matnli ma'lumotlardagi keraksiz bo'shliqlarni, formatlash xatolarini yoki standart bo'lmagan belgilar kodlarini olib tashlashni ham o'z ichiga oladi, bu esa keyingi qayta ishlash bosqichlarining to'g'ri ishlashini kafolatlaydi.

Ma'lumotlarni olish va tekshirish jarayonlarida xatolar yuzaga kelishi muqarrar. Tizimning mustahkamligi xatolarni samarali boshqarish mexanizmiga bog'liqdir. Agar ma'lumot verifikatsiyadan o'tmasa, dasturiy tizim xato holatini aniq tutishi va bu haqda foydalanuvchiga tushunarli qaytma aloqa (feedback) berishi shart. Bu qaytma aloqa foydalanuvchiga qaysi maydonda va qanday xato yuz berganligini ko'rsatib, ma'lumotlarni tuzatishga imkon beradi. Dasturiy me'morchilik nuqtai nazaridan, bu xatolar maxsus istisno sinflari (Exception Classes) orqali boshqarilishi va tizimning asosiy jarayonlarining to'xtab qolishini oldini olishi zarur.

Foydalanuvchidan olingan va verifikatsiyadan o'tgan ma'lumotlarni qayta ishlash murakkab tizimiy vazifadir. Qayta ishlash tizimining kengaytiriluvchanligi (scalability) va parvarishlanishi (maintainability) asosan modulli dizayn tamoyillarini qanchalik to'g'ri qo'llanilganiga bog'liq.

Modulli dasturlash printsipi butun dasturiy tizimni kichik, mustaqil va yaxshi aniqlangan funksional qismlarga, ya'ni modullarga ajratishni talab qiladi. Ma'lumotlarni qayta ishlashda bu yondashuv quyidagi imkoniyatlarni beradi:

Vazifaning Ajratilishi: Har bir modul faqat bitta aniq vazifani bajaradi (masalan, ma'lumotlarni formatlash, hisoblash, ma'lumotlar bazasiga yozish).

Izolyatsiya: Bir moduldagi xato butun tizimning ishdan chiqishiga olib kelmaydi.

Qayta Ishlatish: Modullarni tizimning turli joylarida yoki boshqa loyihalarda takroran ishlatish mumkin.

Bu yondashuv tizimning murakkabligini boshqarishni osonlashtiradi va jamoalararo ishlash samaradorligini oshiradi.

So'nggi yillarda katta hajmdagi ma'lumotlarni qayta ishlashda mikro servis me'morchiligi modulli dizaynning eng ilg'or shakli sifatida keng tarqaldi. Mikro servis yondashuvida butun ilova bir nechta mustaqil xizmatlarga ajratiladi. Har bir xizmat ma'lum bir biznes funksiyasini bajaradi va o'zining ma'lumotlar bazasiga, shuningdek, o'zining texnologiya stekiga ega bo'lishi mumkin.

Foydalanuvchi ma'lumotlarini qayta ishlashda mikro servislari quyidagi ustunliklarni beradi:

Mustaqil Joylashtirish: Har bir xizmatni qolganlaridan mustaqil ravishda joylashtirish (deploy) va yangilash mumkin.

Horizontal Kengaytiriluvchanlik: Tizimning qaysi qismi ko'proq yuklama ostida bo'lsa, faqatgina o'sha mikroservisning nusxalarini qo'shish orqali yuklamani teng taqsimlash mumkin.

Texnologik Moslashuvchanlik: Har bir xizmat o'z vazifasini eng samarali bajaradigan dasturlash tili va freymvorkda yozilishi mumkin.

Mikroservislar ma'lumotlar oqimini kichik, boshqariladigan parchalarga bo'lish orqali tizimning umumiy barqarorligini va kengaytiriluvchanligini dramatik ravishda oshiradi.

Modulli yoki mikroservis me'morchiligida turli xizmatlar orasidagi ma'lumotlar uzatilishini samarali boshqarish eng muhim muammodir. Bu vazifa ko'pincha xabarlar navbati (Message Queues) yoki xabarlar vositachilari (Message Brokers) kabi mexanizmlar yordamida amalga oshiriladi.

Navbatlar quyidagi vazifani bajaradi: foydalanuvchidan olingan ma'lumotlar birinchi (oluvchi) xizmat tomonidan qayta ishlash uchun navbatga asinxron tarzda joylanadi. Keyin boshqa (qayta ishlovchi) xizmat navbatdan ma'lumotlarni o'zi qabul qila oladigan tezlikda olib, ularni qayta ishlaydi. Bu yondashuv ikki asosiy muammoni hal qiladi:

1. Yuklamani Tenglash: Agar ma'lumotlar kirish tezligi qayta ishlash tezligidan yuqori bo'lsa, navbat ma'lumotlarni to'plab, qayta ishlovchi xizmatning cho'kib ketishini oldini oladi.
2. Ishonchlilik: Agar qayta ishlovchi xizmat vaqtincha ishdan chiqsa, ma'lumotlar navbatda saqlanib turadi va xizmat tiklangandan so'ng qayta ishlash davom etadi.

Bu tizim modullar orasida birikmaning past darajasini (low coupling) ta'minlaydi, bu esa tizimning parvarishlanishi va mustahkamligini oshiradi.

Ma'lumotlarni Saqlash va Chiqarish Strategiyalari

Ma'lumotlar qayta ishlash tsiklining yakuniy bosqichi ularni saqlash va foydalanuvchiga natijalarni taqdim etish usulini belgilashdan iborat. Saqlash strategiyasi va natijalarni taqdim etish usuli dasturiy tizimning keyingi ma'lumotlardan foydalanish tezligiga va umumiy foydalanuvchi tajribasiga bevosita ta'sir qiladi.

Qayta ishlangan ma'lumotlarni doimiy saqlash uchun turli xil ma'lumotlar bazasi (turlari mavjud bo'lib, ular turli me'moriy va ishlash talablariga javob beradi.

Relyatsion Ma'lumotlar Bazalari (SQL): Jadvalga asoslangan, qat'iy sxema va tranzaksiya ishonchliligi (ACID) talab qilinadigan ma'lumotlar uchun mos keladi.

NoRelyatsion Ma'lumotlar Bazalari (NoSQL): Yuqori kengaytiriluvchanlik (horizontal scaling), sxemaning moslashuvchanligi va katta hajmdagi ma'lumotlarning tezkor o'qilishini ta'minlashda ustunlikka ega. Bu ma'lumotlar turi, ayniqsa, ma'lumotlar tuzilmasi doimiy o'zgarib turadigan yoki yuqori o'qish/yozish tezligi talab qilinadigan tizimlarda tanlanadi.

Graph Ma'lumotlar Bazalari: Murakkab bog'lanishlar va munosabatlar ustida ishlash uchun, masalan, ijtimoiy tarmoq ma'lumotlarini tahlil qilishda optimal hisoblanadi.

MB tanlovi qayta ishlangan ma'lumotlarning tuzilmasiga va ulardan keyingi foydalanish ehtiyojlariga asoslanishi kerak.

Keshlashtirishning Samaradorligi

Ma'lumotlarni qayta ishlash va saqlash jarayonida keshlash mexanizmi tizimning o'qish tezligini oshirishning eng samarali usulidir. Kesh tizimi, bu qisqa vaqt ichida ko'p marta so'raladigan ma'lumotlarni tezkor xotirada (masalan, RAMda joylashgan Kesh serverlarida) saqlashni anglatadi. Keshga joylashtirishdan maqsad, har bir so'rovni ma'lumotlar bazasiga yuborishdan ko'ra, tezkor xotiradan javob berish orqali javob berish vaqtini (latency) keskin qisqartirishdir. Bu ayniqsa, qayta ishlangan natijalari kamdan-kam o'zgaradigan yoki bir vaqtning o'zida ko'plab foydalanuvchilar tomonidan so'raladigan ma'lumotlar uchun juda muhimdir.

Xulosa

Ushbu ilmiy maqolada foydalanuvchidan ma'lumotlarni olishdan boshlab, ularni qayta ishlash va joylashtirishgacha bo'lgan butun dasturiy ta'minot tsiklining me'moriy yondashuvlari chuqur tahlil qilindi. Biz ma'lumotlar sifatini ta'minlash, tizim barqarorligini oshirish va kengaytiriluvchanlikka erishish uchun zarur bo'lgan ilmiy asoslarni ko'rib chiqdik.

Asinxronlikning Ahamiyati: Ma'lumotlar olish tizimlarida yuqori yuklamani boshqarish va bloklanishni bartaraf etish uchun sinxron yondashuvdan asinxron modelga o'tish me'moriy jihatdan eng maqbul yechim hisoblanadi.

Ikki Bosqichli Himoya: Ma'lumotlarning ishonchliligi va xavfsizligi verifikatsiya (to'g'rilikni tekshirish) va sanitariya (xavfsizlikni tozalash) jarayonlari orqali ta'minlanishi zarur, bu esa tizimni ma'lumotlar butunligi buzilishidan va tashqi hujumlardan himoyalaydi.

Modulli Dizaynning Ustunligi: Qayta ishlash tizimining mikroservislar kabi modulli me'morchilik asosida loyihalashtirilishi tizimning parvarishlanishi, mustaqil kengaytiriluvchanligi va yuklamalarni samarali boshqarishini kafolatlaydi.

FOYDALANILGAN ADABIYOTLAR

1. **Gamma E, Helm R, Johnson R, Vlissides J.** Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
2. **Fowler M.** Patterns of Enterprise Application Architecture. Addison-Wesley, 2003.
3. **Newman S.** Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2014.
4. **Vernon G.** Implementing Domain-Driven Design. Addison-Wesley, 2013.
5. **Cachin C, Guerraoui R, Kapoors B.** Reliable Distributed Systems: Technologies, Web Services, and Future Directions. Springer, 2009.
6. **Saltzer J, Kaashoek M.** Principles of Computer System Design: An Introduction. Morgan Kaufmann Publishers, 2004.