

DECEMBAR 2022. - JANUAR 2023.

PROJEKAT IZ PREDMETA ELEMENTI RAZVOJA SOFTVERA - DOKUMENTACIJA

TATJANA KOSIĆ

PR 33/2020

SANJA JOVIŠIĆ

PR 25/2020

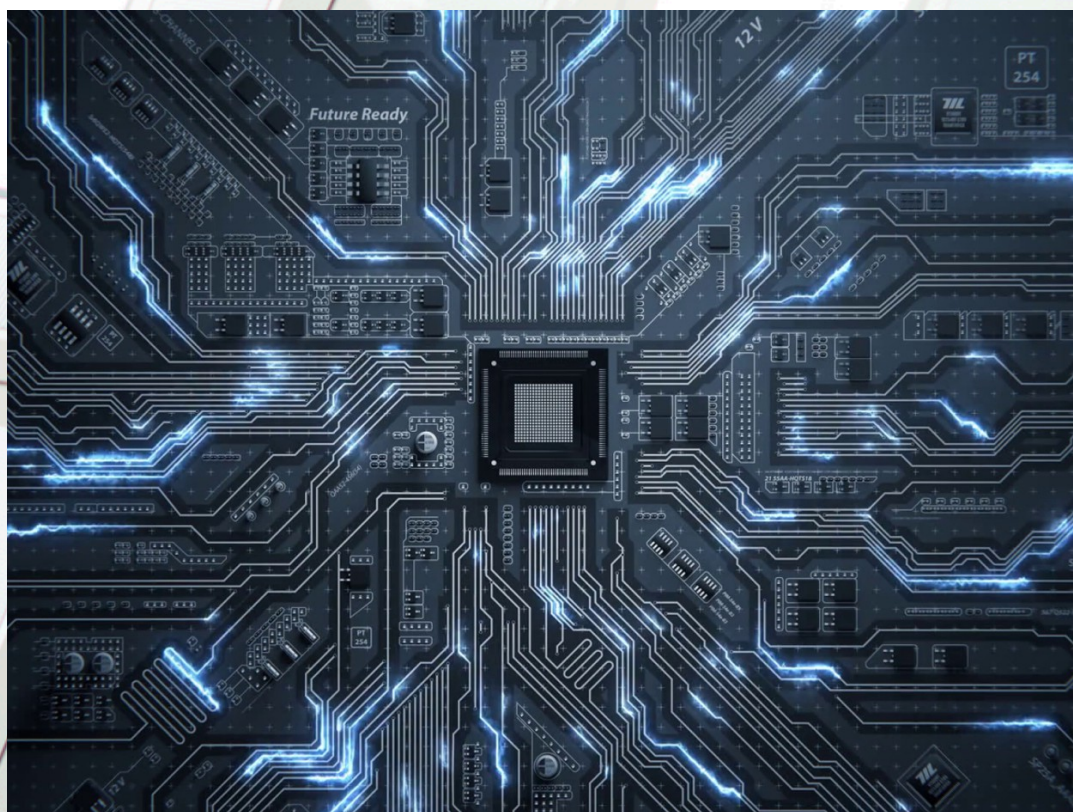
NEVENA MARJANOVIĆ

PR 30/2020

ANJA SABOJLOVIĆ

PR 27/2020

CACHE MEMORY



OPŠTE INFORMACIJE O PROJEKTU

Cache Memory je projekat iz predmeta Elementi razvoja softvera koji se sluša u 5. semestru na Fakultetu tehničkih nauka u Novom Sadu.

Aplikacija prikuplja podatke od korisnika o trenutnoj potrošnji toplotne energije ili o potrošnji u nekom od prethodnih perioda.

KORIŠĆENE TEHNOLOGIJE I OSNOVNA IDEJA PROJEKTA

Tehnologije koje su korišćene prilikom izrade projekta:

- Okruženje: Visual Studio 2019
- Programski jezik: C# - .NET Framework (Console Application, Class Library)
- Testiranje: Unit testovi - Paketi NUnit, NUnit TestAdapter, Moq

Za potrebe razumevanja rada programa, od korisnika se očekuje da ima elementarno znanje vezano za C# i rukovanje okruženjem.

Šta je **Unit testiranje**?

JUnit testiranje predstavlja razvojni proces u kojem se najmanje jedinice aplikacije, koje se mogu testirati, pojedinačno i nezavisno detaljno proveravaju u zavisnosti od željenog ponašanja.

- Junit testovi proveravaju ponašanje jedinice, koja se testira, samo u odnosu na slučajeve od interesa.
- Kada se jednom utvrde svi unit testovi u programu, tako da rade na efikasan i pravilan način (bez grešaka), može se preći na proveravanje većih komponenti.

KONTEKST PROBLEMA

Svakodnevnica nam donosi različite poteškoće po pitanju sakupljanja ogromne količine podataka o potrošnji toplotne energije. Broj korisnika stalno raste, te se javlja potreba da se neki rutinski poslovi automatizuju. Problem se javlja kod beleženja potrošnje toplotne energije kada se podaci unose pojedinačno gde ne postoji nikakva veza sa prethodno unetim podacima. Takvi podaci nisu od preteranog značaja za firmu koja se bavi distribucijom toplotne energije, jer ne može da formira statistički pregled podataka i optimizovati distribuciju tako da svi korisnici budu zadovoljni.

REŠENJE PROBLEMA

Problem se može rešiti formiranjem jedinstvenog **informacionog sistema** (IS) koji će opsluživati aplikacija preko koje će korisnici unositi podatke o potrošnji toplotne energije. Na taj način se omogućava uvid u statistiku potrošnje, ali i filtriranje podataka prema željenim kriterijumima.

Da bi se ovo rešenje moglo sprovesti potrebno je uvesti **mikroservise** odnosno komponente koje su međusobno povezane i komuniciraju jedna sa drugom.

KOMPONENTE APLIKACIJE

WRITER KOMPONENTA

Mikroservis koji prima podatke od korisnika, ali ih ne zadržava za sebe jer nema svoju memorijsku strukturu. On nije zadužen za proveru ispravnosti podataka i zato ga ne zanima kakvi su podaci koje prima. Njegov osnovni zadatak je da informacije unete od strane korisnika prosledi sledećem sloju za obradu.

DUMPING BUFFER KOMPONENTA

Mirkoservis koji privremeno čuva podatke koji su mu prosledjeni, tj. kreira red čekanja i omogućuje dodavanja podataka u red, i periodičnu proveru popunjenosti reda čekanja. Ako je ispunjen uslov popunjenosti, podatke prosledjuje sledećem sloju za obradu.

HISTORICAL KOMPONENTA

Najvažniji mikroservis koji komunicira sa izlaznim fajlom tako sto upisuje podatke u njega. On jedini poseduje interaktivni režim rada jer ima uvid u informacije koje je korisnik prvobitno uneo.

READER KOMPONENTA

Mikroservis koji je posrednik izmedju komponenti informacionog sistema ima ulogu da vizualizatoru podataka vrati iščitane podatke iz fajla. Kao mikroservis koji čita podatke, on takodje ne vodi brigu o poznavanju podataka, već samo vraća podatke koji su filtrirani na osnovu korisničkih kriterijuma, kojih može biti više. Kao rezultat, korisniku može biti prikazan jedan podatak(jer jer zadovoljen samo jedan kriterijum), listu podataka (u slučaju da više podataka zadovoljava kriterijum) ili praznu listu (ako nijedan podataka ne zadovoljava kriterijum).

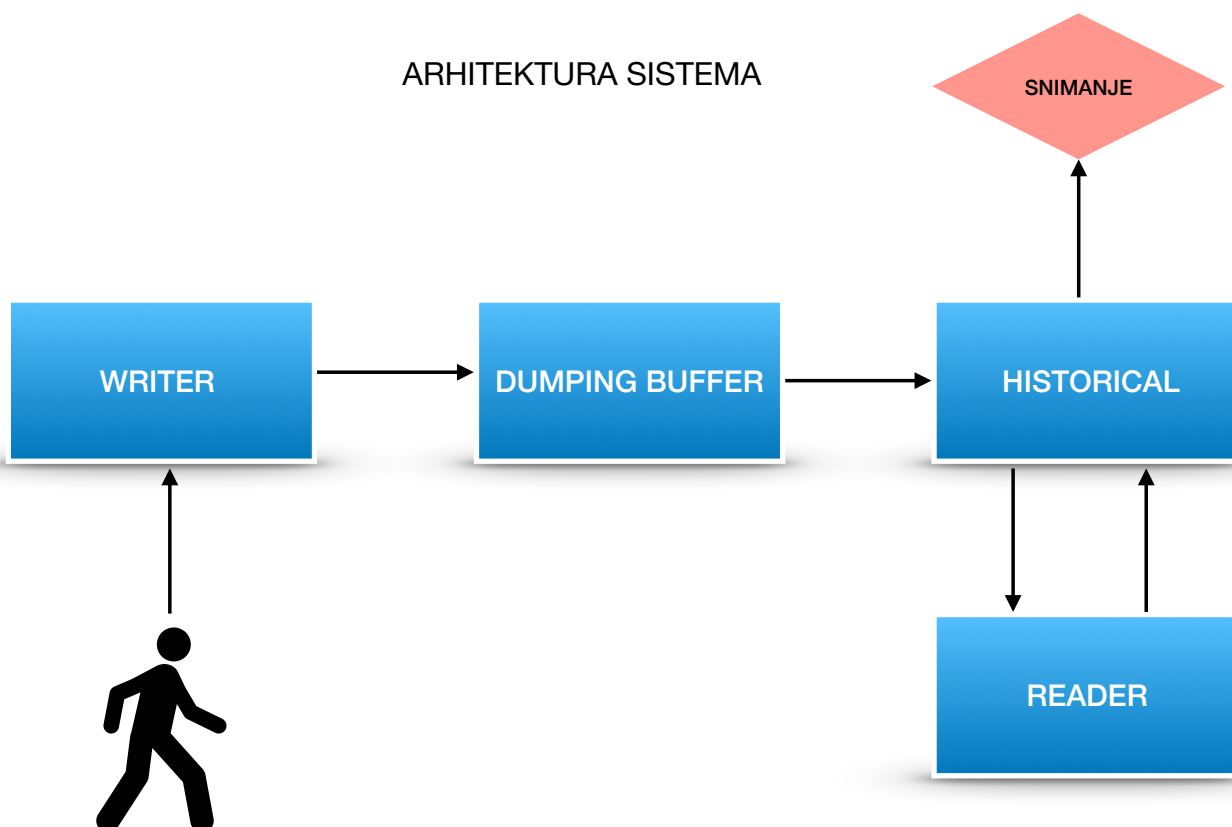
CILJEVI PROJEKTA

1. Mogućnost unosa podataka o korisniku/brojilu
2. Mogućnost unosa podataka o potrošnji
3. Mogućnost pregleda postojećih i unetih podataka
4. Mogućnost komunikacije komponenti u unutrašnjosti sistema
5. Mogućnost pregleda podataka po kriterijumima
6. Repetitivni unos podataka

ARHITEKTURA SISTEMA

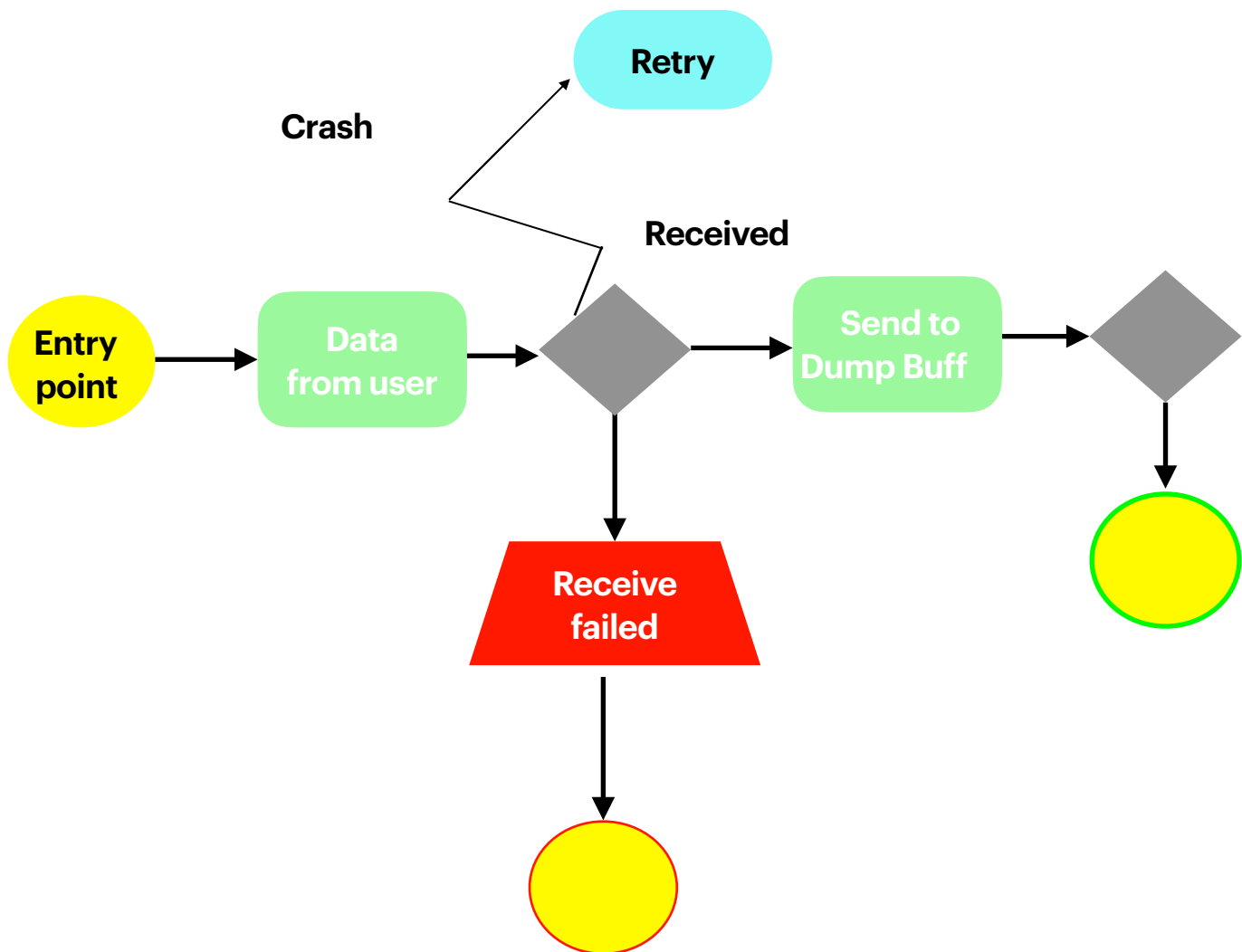
Dizajn arhitekture sistema opisuje i određuje sistemski i aplikativni softver, hardver i mrežno okruženje sistema. Formira se specifikacija koja treba da opiše osnovne komponente informacionog sistema, arhitekturu sistema i specifikacije softvera i hardvera. Specifikacija se formira na osnovu funkcionalne i systemske specifikacije zahteva korisnika, modela procesa, modela podataka, biznis proces modela. Pri izradi specifikacije mora se voditi računa o operacionim zahtevima (operativni sistem), performansama sistema (brzina, količina podataka) i bezbednosnim zahtevima (korisnici, šifre, prava, virusi).

ARHITEKTURA SISTEMA



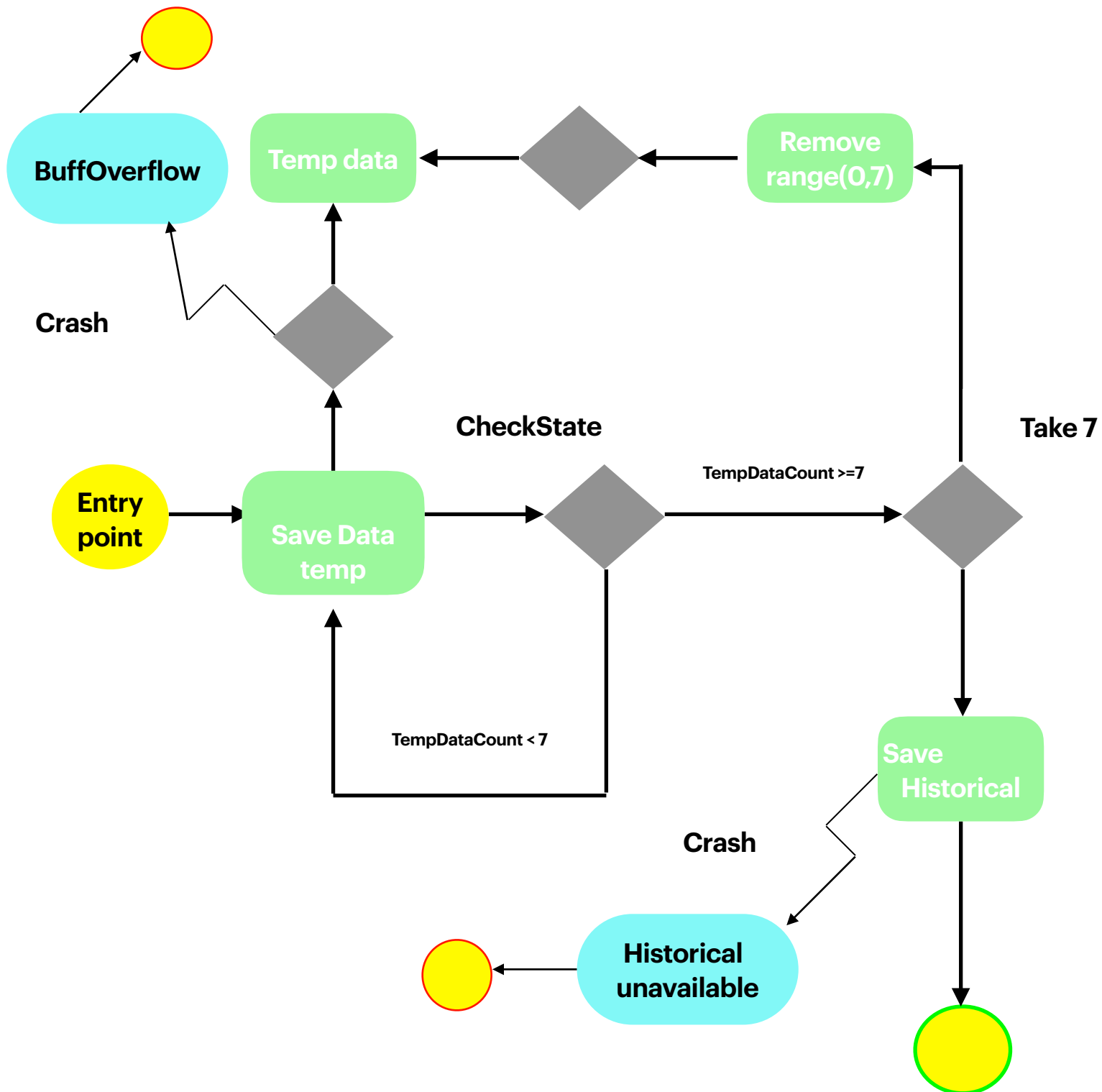
DIJAGRAMI AKTIVNOSTI

DIJAGRAM WRITER KOMPONENTE



Writer komponenta prima podatke koje je korisnik uneo. U zavisnosti od uspešnosti usnosa, podatke šalje sledecoj komponenti ili zahteva novi unos od korisnika ukoliko je uneo podatke u pogrešnom formatu(crash grana).

DIAGRAM DUMPING BUFFER KOMPONENTE



Dumping Buffer prima podatke od komponente Writer I čuva ih u privremenu listu podataka. U zavisnosti od toga da li je u listu podataka upisano manje ili više od 7 podataka, vraćamo se na čuvanje podataka ili idemo na sledeći korak. Uzima se 7 elemenata iz liste i prosledjuje Historical komponenti (ukoliko je ona dostupna, ako nije, izlazimo iz programa).

PRIKAZ KONZOLE

Unesi id:

1

Unesi adresu:

Glavna ulica

Unesi mesec:

jan

Unesi potrošnju:

45

Dodato u red...

Prikaz podataka? da/ne

da

Citanje po kriterijumu?

1 - Po mesecu

2 - Po adresi

3 - Po korisniku

4 - Ispis svih

ZAVRŠNA NAPOMENA

Namena ovog teksta je da korisniku prvenstveno što više pojaśni i predstavi način na koji aplikacija funkcioniše kako bi se koristila na ispravan način.