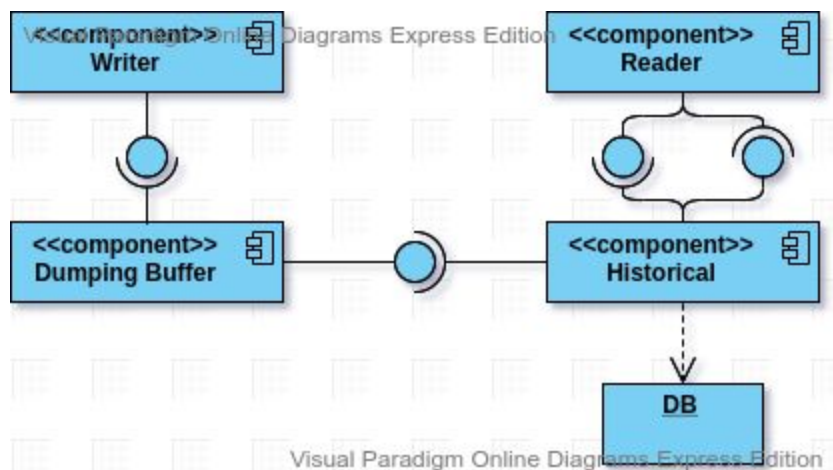


## Cache Memory

Потребно је направити дизајн система, архитектуру система, имплементира и истестирати решење који симулира рад и комуникацију кеш меморије са перзистенцијом података.



### Начин рада система

Систем садржи 4 компоненте:

1. Dumping Buffer
2. Historical
3. Writer
4. Reader

**Writer** компонента прослеђује податке на 2 начина:

Податке шаље на сваке 2 секунди Dumping Bufferкомпоненти (WriteToDumpingBuffer)

Податке шаље директно за складиштење Historical компоненнти (ManualWriteToHistory)

Dumping Buffer компонента добијене податке препакује у своју структуру са којом рукује, при томе додељује јединствен ИД. Датасет се додељује према Code-у података који су примљени.

Dumping Buffer компонента у својој структури складишти податке у меморији (CD структура).

Уколико од Writer компоненте пристигну подаци који имају исти Code као и они што се налазе у Dumping Buffer компоненти, њихова вредност (Value) у Dumping Buffer компоненти треба да се ажурира на нову вредност која је стигла од Writer-а.

Кад Dumping Buffery својој колекцији накупи 2 различите вредности у оквиру истог датасет-а, тада су подаци спремни за слање и пакују се у DeltaCD компоненту, затим се ослобађа CD

структура за пријем нових елемената. После пристиглих 10 вредности из Writer компоненте DeltaCD се прослеђује Historical компоненти (WriteToHistory). Том приликом уколико DeltaCD нема ниједан од Collection Description објеката (Add, Update или Remove) слање ће се одложити за још 10 пријема података од Writer компоненте. Уколико пре слања DeltaCD компоненте, Dumping Buffer накупи опет 2 различите вредности у оквиру истог датасет-а, тада ће ти подаци чекати, и уколико је потребно ажурирати се са Writer-а док се не пошаље DeltaCD компонента и тада је могуће уписати нове елементе.

За потребе DeltaCD компоненте потребно је водити рачуна који елементи су нови и стижу за Add, који за Update, а који за Remove у Dumping Buffer-у. Из разлога што Writer има могућност мануелног уписа у историју, Dumping Buffer у неким случајевима не може да зна да ли постоји елемент у историји и при томе Historical треба да чува информацију да је податак стигао преко мануелног уписа, и да ће једино валидно бити да се ажурира преко CollectionDescription Add из Dumping Buffer-а, док није валидно да Dumping Buffer пошаље Update и Remove за тај елемент јер нема информацију да ли је уписан.

Historical компонента прима садржај од Dumping Buffer компоненте и препакује структуру од Dumping Buffer компоненте, DeltaCD, у своју интерну структуру – LD, том приликом се мапира DumpingValue на HistoricalValue.

Historical компонента треба да провери да ли су подаци валидни – да ли су датасет-ови одговарајући и у складу са Code-овима који су прослеђени у оквиру датасет-а.

Historical компонента треба да провери да ли добијени подаци треба да буду уписану у базу података. Услов да се податак упише у базу података је да излази из Deadband-а.

Deadband представља услов да ли је потребно податак уписати у базу података.

Deadband износи 2% и значиће да уколико пристигли податак, већ постоји у бази, уколико нова његова вредност је већа од 2% од старе вредности, тада ће бити уписана нова вредност. Уколико нова вредност не излази из оквира од 2% од старе вредности тада нова вредност не треба да буде уписана у базу података.

Једини изузетак из Deadband-а је Code – CODE\_DIGITAL, за овај Code се увек уписује прослеђена вредност и не проверава се Deadband.

Приликом уписа податка у базу, Historical компонента ће генерисати тиместамп са временом уписа тог податка.

Reader компонента треба да ишчита вредности из Historical компоненти по временском интервалу за тражени Code (GetChangesForInterval).

Имплементирати Logger, који ће бележити све активности које се дешавају у осталим компонентама.

# Модел података

## Dumping Buffer

Dumping Buffer је компонента која служи за привремено чување података пре него што их проследи Historical компоненти. Dumping Buffer прима податке од компоненте Writer, од које добија све податке.

Dumping Buffer чува податке у облику колекције – CollectionDescription (CD).

CollectionDescription садржи:

- ID
- Dataset
- DumpingPropertyCollection

DumpingPropertyCollection садржи:

- низ DumpingProperty-a

DumpingProperty:

- Code
- DumpingValue

Приликом слања података Historical компоненти, Dumping Buffer креира посебну структуру DeltaCD.

- Transaction ID
- CollectionDescription Add
- CollectionDescription Update
- CollectionDescription Remove

DeltaCD садржи 3 CollectionDescription структуре, по једну за операције додавања, ажурирања и брисања из складишта.

## Historical

Historical је компонента која служи за перзистенцију података добијених од Dumping Buffer компоненте.

Historical компонента податке добијене од Dumping Buffer компоненте снима у базу података.

Снимање у базу података се врши према датасет-у који је добијен као податак од Dumping Buffer компоненте. За сваки датасет треба да постоји посебна табела који ће чувати измене.

Приликом перзистенције података, Historical компонента ради са својим структурама:

HistoricalProperty:

- Code
- HistoricalValue

Description:

- ID
- листа HistoricalProperty-s

- Dataset

Lista Description-a (LD) sadrži:

- листу Description-a

Historical компонента ради са LD структуром и то складишти.

## Writer

Writer је компонента која служи за уписивање нових података у Dumping Buffer. Writer компонента прима нове информације сваке 2 секунде и то прослеђује Dumping Buffer -у. Writer има и могућност директног уписа у историју, преко Historical компоненте. Writer компонента нема своју меморијску структуру са којом ради, већ податке директно шаље Dumping Buffer -у. Подаци који се шаљу су:

- Code
- Value

## Reader

Reader је компонента која служи за читање података из базе података. Reader комуницира са Historical компонентом ради добављања потребних података и исписа тих података.

## Додатне информације

Листа кодова:

1. CODE\_ANALOG
2. CODE\_DIGITAL
3. CODE\_CUSTOM
4. CODE\_LIMITSET
5. CODE\_SINGLENODE
6. CODE\_MULTIPLENODE
7. CODE\_CONSUMER
8. CODE\_SOURCE
9. CODE\_MOTION
10. CODE\_SENSOR

Списак скупова податак по кодовима:

- DataSet = 1 – CODE\_ANALOG, CODE\_DIGITAL
- DataSet = 2 – CODE\_CUSTOM, CODE\_LIMITSET
- DataSet = 3 – CODE\_SINGLENODE, CODE\_MULTIPLENODE
- DataSet = 4 – CODE\_CONSUMER, CODE\_SOURCE
- DataSet = 5 – CODE\_MOTION, CODE\_SENSOR
- 

\*Value има структуру:

- Timestamp (датум и време)
- ID географског подручја
- Потрошња и mW/h

## Критеријум оцењивања

1. Дизајн и архитектура решења
2. Коришћење *Scrum* методологије развоја – дефинисање *user story*-а и таскова, планирање и естимација
3. Имплементација решења
4. CI циклус
  - a. Build
  - b. Unit Tests
  - c. Покривеност кода тестовима