

Analiza unui sistem de monitorizare și control al nivelului de apă dintr-un bazin

1.a) Cerințe funcționale

- Programarea celor 4 pompe să permită alimentarea bazinului până la limita maximă
- Nivelul de apă din bazin trebuie menținut între o limită minimă și una maximă reprezentată de doi senzori
- Pornirea semnalului sonor în momentul depășirii limitei superioare și deschiderea robinetului de evacuare a apei din bazin.
- Crearea unei interfețe prin care să poată fi vizualizată starea sistemului în timp real
- Adăugarea unor butoane care să permită oprirea pompelor de alimentare de către utilizator
- Stocare sesiunilor de lucru într-o bază de date care să permită redarea comportamentului sistemului din anumite momente
- Adăugarea unui ecran de vizualizare a istoricului de funcționare a sistemului

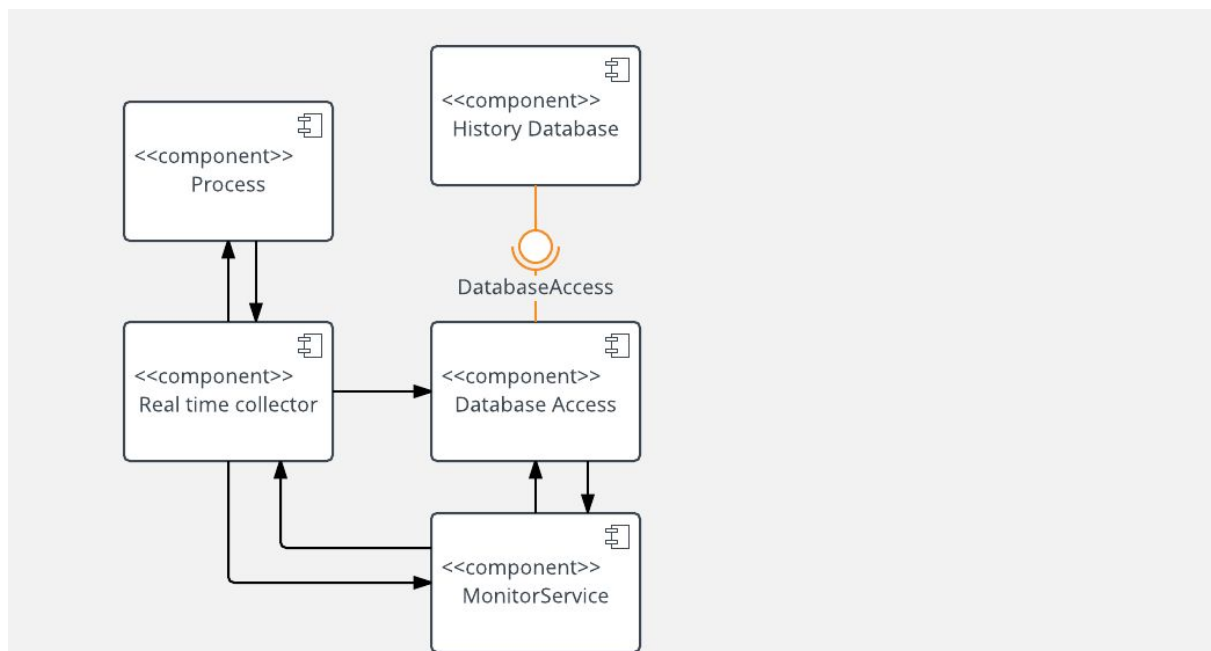
1.b) Cerințe non-funcționale

- Timpul de răspuns al pompelor față de comenzi din interfață să fie minim
- Protocolul de comunicare între aplicația de monitorizare și automat să fie TCP(Transfer Control Protocol)
- Socketul să fie închis la încheierea fiecărei sesiuni de monitorizare
- Să se folosească o aplicație de tip WPF(Windows Presentation Foundation) pentru implementarea aplicației de monitorizare
- Se va folosi un sistem de logare a evenimentelor care au loc în sistem care să permită identificarea rapidă a eventualelor probleme
- Aplicația de monitorizare să implementeze pattern-ul MVVM(ModelView-ViewModel)

1.c) Cerințe care influențează arhitectura

- Stocare stării sesiunilor într-o bază de date

2.a) Diagrama de componente



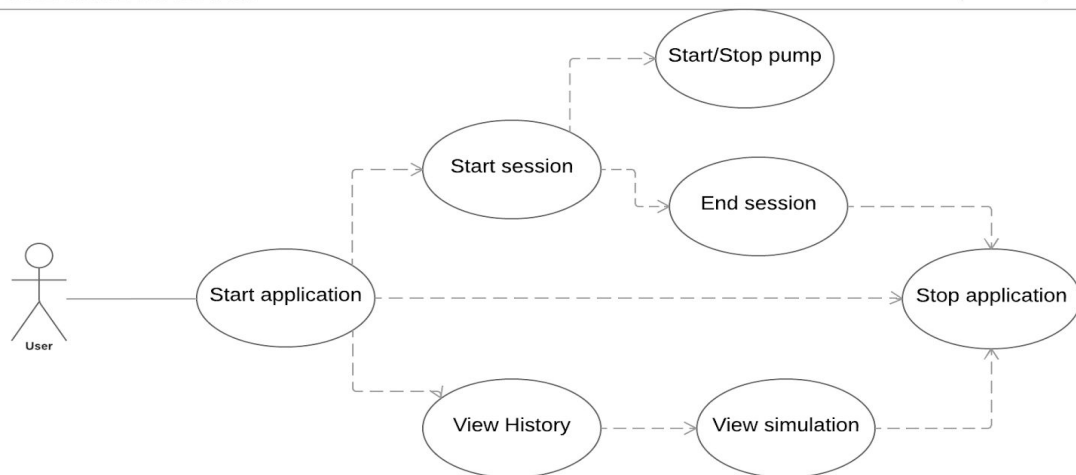
- Componenta proces este responsabilă de simularea nivelului de apă din bazin prin pornirea/oprirea pompelor de alimentare. Aceasta comunică cu componenta Real Time Collector care trimite date despre starea procesului către componenta Monitor Service
- Monitor Service este componenta care afișează informații despre starea procesului
- Database access este componenta care asigură scrierea și citirea datelor necesare simulării unei sesiuni

Principalele componente ale sistemului sunt Proces și Aplicație de monitorizare care la rândul ei este împărțită în componente după cum urmează: Real time collector: colectează date de la proces pentru construirea sesiunii simulate în aplicația de monitorizare. Acesta comunică cu Monitor service și Database access pentru stocarea informației referitoare la sesiune. History database este componenta care stochează datele referitoare la istoricul sesiunilor.

3. a) Diagrama use case:

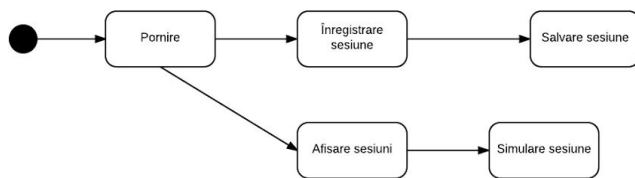
BASIC USE CASE DIAGRAM

Andrei Prodaniuc | October 1, 2017



Pentru a putea începe procesul de monitorizare utilizatorul trebuie să pornească aplicația. Odată pornită aplicația, utilizatorul poate fie să pornească o nouă sesiune de monitorizare, fie să vizualizeze istoricul sesiunilor. În cazul pornirii unei noi sesiuni utilizatorul poate să acționeze asupra pompelor (pornire/oprire) și, de asemenea, să oprească respectiva sesiune. În cazul vizualizării istoricului se poate trece mai apoi la simularea unei sesiuni din baza de date. Stop application este pasul închiderii aplicației.

3. b) Diagrama de stări:



4.a) Modele arhitecturale folosite

Pentru realizarea aplicației de monitorizare a procesului s-a ales folosirea modelului arhitectural MVVM (ModelView-ViewModel). S-a ales folosirea acestui model în primul rând pentru modul în care pot fi separate sarcinile. Modelul este componenta care ne dictează cum anume arată datele care trebuie reprezentate. ViewModelul se ocupă de procesarea datelor și de trimiterea lor în forma potrivită către View care afișează datele primite. În acest model arhitectural este important modul în care comunică ViewModelul și View-ul, lăsând impresia unui ui responsive și user-friendly.

5.a) Indicatori de calitate

Performanța: sistemul fiind unul cu monitorizare în timp real, performanța este unul dintre indicatorii indispensabili. Timpul de răspuns este măsura care definește acest indicator, acesta fiind de ordinul milisecundelor (de ex: timpul dintre apăsarea unui buton și oprirea pompei respective)

Toleranța la modificări: Sistemul este deschis la adăugarea de noi componente. Aplicația de monitorizare fiind o aplicație desktop poate cu ușurință să accepte comunicarea cu o aplicație web, sau poate introduce notificări care să semnaleze utilizatorul

în cazul depășirii unor anumite valori prestabilite. Costul implicat în aducerea de astfel de modificări sunt relativ reduse.

Disponibilitatea: Sistemul este proiectat în așa fel încât să ofere o disponibilitate de 24 ore/24. Aplicația poate fi instalată pe mai multe computere, ceea ce ar duce la menținerea stării de monitorizare în cazul în care un computer devine indisponibil.

6.a) Tehnologii middleware

În procesul de comunicare dintre aplicația de monitorizare și simulatorul procesului se folosește comunicarea de tip client-server. Între cele 2 componente se deschide un socket și, prin protocolul TCP se transmit date despre starea sistemului care sunt afișate în interfață.

7.a) Validarea arhitecturii

Pentru validarea arhitecturii se consideră următoarele scenarii:

- pornirea sistemului: o nouă sesiune de înregistrare a datelor este pornită, în acest moment datele referitoare la fiecare stare a sistemului din respectiva sesiune sunt serializate în vederea salvării în baza de date
- redarea unei sesiuni: datele despre stările din care este compusă sesiunea sunt deserializate și mai apoi mapate peste interfața grafică care simulează sesiunea