

En beräkningsövervakare

Det här är Computell

- Visar status f\u00f6r dina ber\u00e4kningar i en grafisk dashboard
- Kan spara loggmeddelanden i csv format för vidare analys
- Låter dig ha koll på distans via e-post
- Fungerar tillsammans med Python och Java

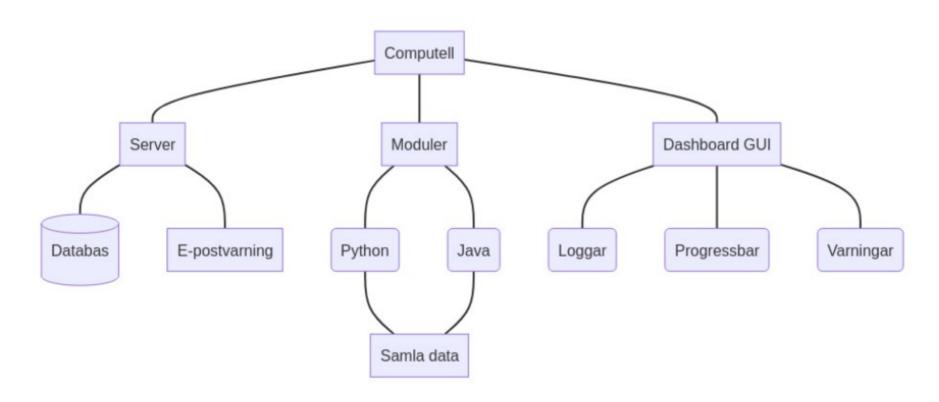
Utvecklat nära kunden

Must have	Should have	Could have	Won't have
Bibliotek för Python	Möjlighet att skicka med rubrik eller annan info från "mätpunkterna"	Exportfunktion av crash och varningar	Dashboard i mobilen
Bibliotek för Java	Progress bar	Prioriterade varningar	
Möjlighet att logga ett meddelande per sekund Möjlighet att köra	Någon typ av loggfunktion med filterfunktion för olika typer av meddelanden	Möjlighet att lägga server och dashboard på andra maskiner än där beräkningarna körs	
beräkningar/server/dashboard på samma dator	Grafisk dashboard eller lokal webbdashboard	Terminalbaserad dashboard	
Server för att samla in data	Möjlighet att skilja på fel och andra statusuppdateringar i dashboard	SMS-notifikation vid crash	
Dashboard			

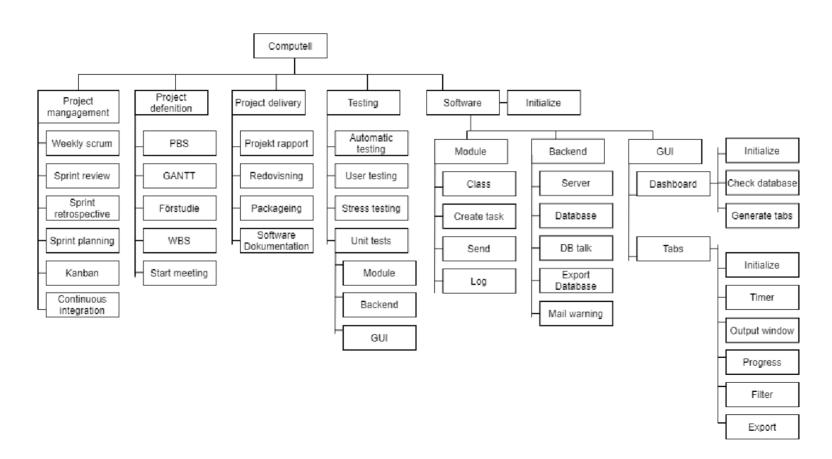
Utvecklat nära kunden

Must have	Should have	Could have	Won't have
Bibliotek för Python	Möjlighet att skicka med rubrik eller annan info från "mätpunkterna"	Möjlighet att lägga server och dashboard på andra maskiner än där beräkningarna körs	Dashboard i mobilen
Bibliotek för Java	Progress bar	Terminalbaserad dashboard	
Möjlighet att logga ett meddelande per sekund	Någon typ av loggfunktion med filterfunktion för olika typer av meddelanden	SMS-notifikation vid crash	
Möjlighet att köra beräkningar/server/dashboard på samma dator	Grafisk dashboard eller lokal webbdashboard		
Server för att samla in data	Exportfunktion av crash och varningar		
Dashboard	Prioriterade varningar		
Möjlighet att skilja på fel och andra statusuppdateringar i dashboard			

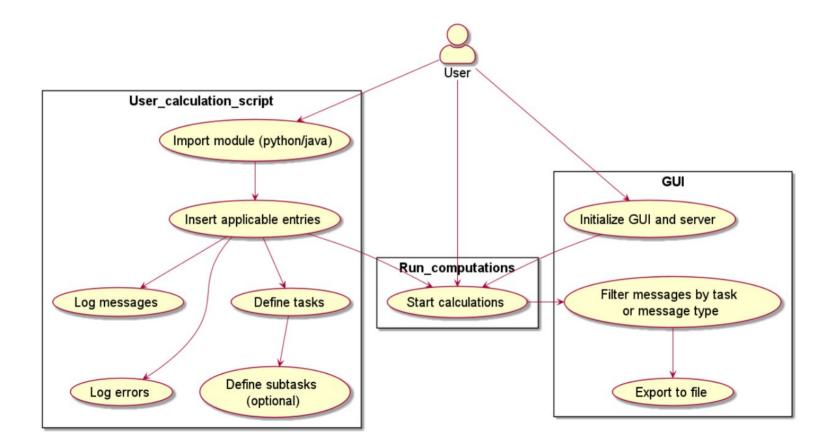
Product breakdown strukture



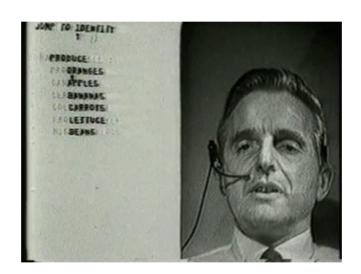
Work breakdown structure



Så används Computell

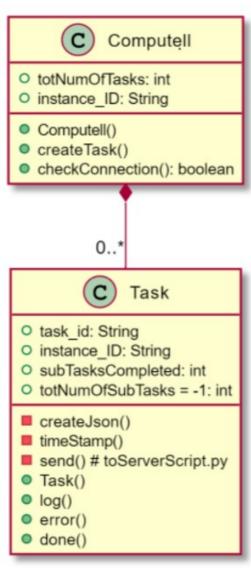


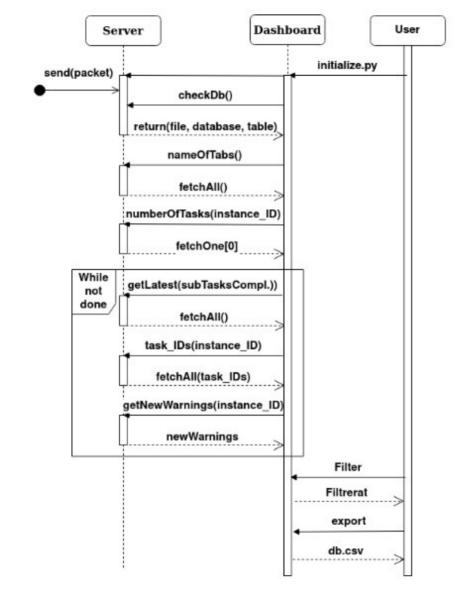
Demo



Så funkar det

```
import computell.*
// Vi vill övervaka 2 beräkningar rörande en stjärna.
Computell sagittarius AStar = new Computell("Sagittarius A-Star", 2)
// Skapar ett task för beräkning av avstånd.
Computell.Task distance = sagittariusAStar.createTask(
                            "Lightyears from earth", 1000)
// Skickar ett log-meddelande.
distance.log("Startar mätning av avstånd ")
for (datapoint : data) { // data.size() = 1000
   // Beräkning
    distance.done()
                       // Uppdaterar "progress".
   if (!datapoint) {
       // Skickar ett fel-meddelande med 'severity' 1.
        distance.error("Saknar en datapunkt, fortsätter beräkning", 1)
Computell.Task volume = sagittariusAStar.createTask("Volume")
volume.log("Startar beräkning av volym")
// *Okänt antal delberäkningar*
   volume.done()
    if (singularity) {
       // Skickar ett fel-meddelande med 'severity' 3.
       volume.error("Singularitet utan volym, avbryt beräkning", 3)
    }
```





Test

- Continuous integration
- Unit tests
- Automatiska test
- Användartest/användbarhetstest

Stresstestning

- Lagringsfrekvens
- Lagringskapacitet

Tack!