Website Availability and Performance Monitoring

Gabrielle Rappaport

2017/12/04

1 Technologies

1.1 JAVA

I decided to develop the console app using Java mostly because I have experience using it and because I think Java is relevant for application development since it is completely object oriented. Java allowed me to develop the application in an object-oriented way and to structure the application using design patterns. Thus, it allows me to have a well-structured and scalable application that is much easier to maintain and more efficient.

1.2 MySQL

I decided to store the data - the different checks made on the websites and the website's data - in a SQL database. I think it is important to have a persistent storage of the checks so we can keep a track on the informations we retrieved. I chose MySQL because it is available on many OS, openSource, it is rather permissive with the data it contains and easy to use. It lacks some functionalities (such as class inheritance) and is less robust on large volumes than PostgreSQL for example, but in the case of a console application, I don't think it is necessary to use a too powerful DBMS. Anyhow, I implemented a DAO that would allow me to easily rotate to another DBMS if necessary.

2 Features and Architecture

See JavaDoc for detailed explanations on the classes.

2.1 Data Tier

I implemented the data tier with the Data Access Object design pattern, you can find it in the dao package. The data tier includes the data persistence mechanisms and the data access layer that encapsulates the persistence mechanisms and exposes the data : the two interfaces CheckDao and WebsiteDao.

Avoiding dependencies on the storage mechanisms allows for updates or changes without the application tier clients being affected by the change which gives the application more scalability and maintainability.

2.2 Application Tier

The application tier is responsible for performing detailed processing : it retrieves the data from the data tier and pushes it to the presentation tier. In the monitor package, you can find the classes that will perform the checks on the website, create the alerts when necessary and retrieve the data from the dao package.

Users can keep the console app running and monitor the websites, that is to say that the user can have the monitoring running while adding, deleting and updating the website database : when a website is added to the database, the checks and the alerts are immediately scheduled, there is no need to relaunch the application to make the change effective. Likewise, when a website is deleted from the database, the checks and alerts stop immediately and the Check database is cleared, even if the application is running.

2.3 Presentation Tier

2.3.1 Command Line User Interface

The User can communicate with the application only through the console : he can add, delete, update websites and run the monitoring. Moreover, the alerts are prompt in the console so the important informations are displayed in a synthetic way. When the user looks at the console, he can only see the methods he called and the alerts that have been raised.

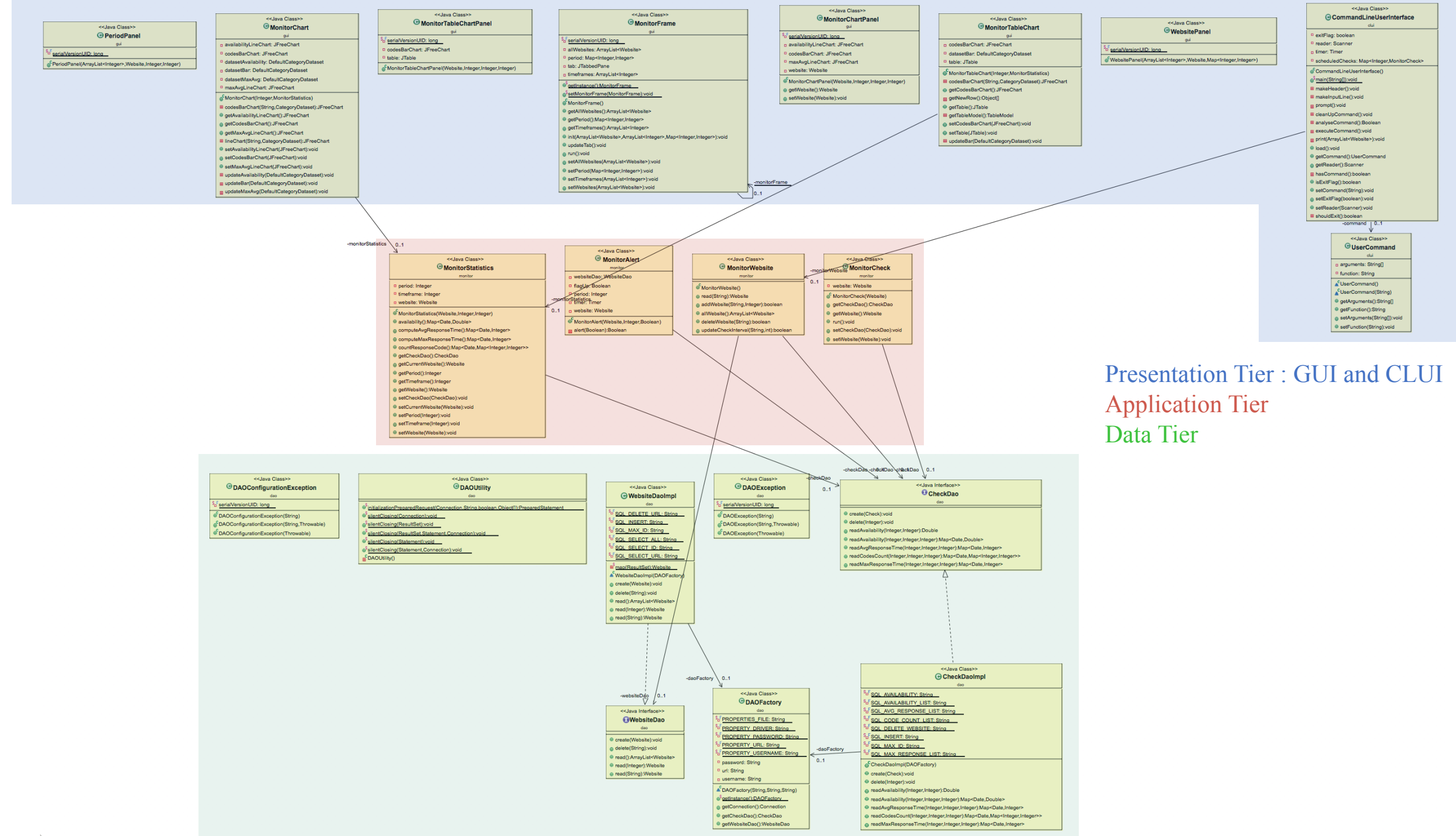
2.3.2 Graphical User Interface

Since it is a monitoring application, I think it is important for the user to be able to visualize the detailed statistics with a graphical interface. In fact, to have a bigger picture of the monitoring, he can look at the console and check on the alerts. However, if the user wants more details on a website, I think it is important to provide a graphical interface on which the statistics are displayed. Thus, I implemented a graphical interface with Java to prompt the statistics.

The GUI prompts the statistics over two different timeframes - 2 and 10 minutes - for two different periods - 10 minutes and 1h. *Note that those variables are easily changeable in the CLUI, so the code is modular.*

3 Application Design Improvement

- I think it would be better to leave the choice of the timeframes (2 and 10 minutes), the period (10 minutes and 1 hour) and the refresh intervals (respectively 10 and 60 seconds) to the user. The user could set those parameters when starting the application, or even better, he can update those anytime when the application is running. The application would have the advantage to be better suited to the user exigences, thus to please a bigger population. Moreover, it means that the parameters could be modular through time and be adapted to the different websites. Finally, it would make the visualization of the data way clearer which would help the user identify the right statistics faster.
- Regarding the alert system, it is better to leave the possibility to the user to choose the limit under which the alert is triggered, according to each website he wants to monitor. It could also be possible to create different alert levels. Indeed, a user can have several websites to monitor with different priorities on each website. Having the possibility to indicate a different level of alert for each websites, or to define an order of priority on the websites can help the user to quickly identify the flaws on the sites that he monitors.
- Other metrics can be left to the user's choice as the timeout - the time to allow for a response.
- It would also be interesting to leave the opportunity to the user to choose which statistics he wants on each website. We could propose him a list of different statistics and give him the possibility to create his own dashboard. In this project, I computed the HTTP Status count, the availability, the maximum and average response time, but we could also think about other metrics like the request rate - how much traffic your application receives, it will have an impact on all the other metrics and will help the user understand how its website scales- or saturation - the amount of requested work that the website has queued.


PeriodPanel
MonitorChart
MonitorTableChartPanel
MonitorFrame
MonitorChartPanel
MonitorTableChart
WebsitePanel
CommandLineUserInterface
UserCommand
MonitorStatistics
MonitorAlert
MonitorWebsite
MonitorCheck
DAOConfigurationException
DAOUtility
WebsiteDaoImpl
DAOException
CheckDao
WebsiteDao
DAOFactory
CheckDaoImpl
Presentation Tier : GUI and CLUI
Application Tier
Data Tier