

Database Performance Dashboard

CS5226 Project Report

Gu Yang A0000000

Lu Fangjian A0040740W

Yang Zhaoyu A0040953J

Zhang Haojun A0000000

**Project Admin**

|  |  |  |
| --- | --- | --- |
| **S/N** | **Task** | **Member\*** |
| 1 | 3 levels of breakdown for each database parameter being monitored. | Lu Fangjian |
| 2 | Provide a configuration page(s) to define the various thresholds used to determine whether a specific database parameter is healthy, not so healthy or need DBA attention. | Lu Fangjian |
| 3 | For each of the performance issue identified that is in the “red” area, the application should provide information on the parameter in init.ora that should be modified in order to solve the performance issue. |  |
| 4 | On-demand reports – where the users can specify the date ranges for the database parameters being monitored. | Lu Fangjian |
| 5 | A Debug interface – Allow the user to issue SQL commands to the database. The results should be displayed neatly on the webpage. |  |
| 6 | README.txt |  |
| 7 | Database Setup Scripts |  |
| 8 | Configure the web development framework (Spring 3.2.0). Setup the project on google code. | Lu Fangjian |
| 9 | UI design and modification. | Lu Fangjian |
| 10 | Configure connection parameters to dbtune server. | Lu Fangjian |
| 11 | Project Report writing. | Gu Yang  Lu Fangjian  Yang Zhaoyu  Zhang Haojun |
| 12 | Additional stuff you have done | Append your names in this column |

\*Members are listed in alphabetic order

**System Design and Architecture**

**The overall architecture and design of the application**

The web application uses the pre-installed Oracle database in dbtune server to perform its main activities, which is to monitor a set of database parameters, and to query database through debug window. The database is also acted as permanent storage for the configuration data.

The development of the web interface follows the popular Model-View-Controller (MVC) approach using Spring Framework 3.2.0. The various application layers are separated distinctly in different classes. The detailed relations are illustrated below.

|  |  |  |
| --- | --- | --- |
| **Class** | **Folder** | **Role** |
| \*TO.java | domain | Transfer objects are data carriers which are used in Presentation layer (web interface), Service layer and DAO layer. |
| \*.jsp | view | Views are web pages which receive transfer objects from controller and display them to user. They also allow user to send inputs to controller for processing. |
| \*Controller.java | controller | Controllers fetch the transfer objects and send them to the view for display. They also receive user inputs and send them to service layer for analysis. |
| \*Service.java | service | Services are implemented with business logics to perform analysis. They may also fetch transfer objects from DAO layer during the process. |
| \*Dao.java | dao | Data access objects retrieve data from permanent storages such as database, put them into transfer objects and send them to service layer for analysis. |
| \*Util.java | util | Constants and static variables/methods which can be utilized to perform certain functions. |