Sistem Analisis Description

Architecture

For best control of the backend functions, he will be divided into two subsistens. First, an Update Server, responsable for accessing the GitHub API, retrieving all the data and input or updating them in the database. Second, a RESTFul server that will perform all the data processing and make them available for the frontend applications.

Database

The project's database is SQLite. A good choice for a relational database to systems that will run locally. The entity model, design and create script of the database will be available in the documentation folder.

updateServer

That service will be responsible for accessing the GitHUb API, collecting, organizing and inserting them in the database, also it'll make the statistics calculation with collected data. An automatic update can be setted to run in an interval of time, by default, setted to run every 12 hours.

Libraries

- ➢ Express: Web Framework;
- ➢ Consign: Gives an easy way to organize your Router Files;
- ➢ Axios: An easy to use HTTP client;
- ➢ Parse-link-header: A parser lib for Link information provided by Headers on GitHub API;
- ➢ Sqlite3: Database lib;

Performance

The GitHub API has a pagination system for all requests, limiting your reading by maximum 100 rows for page request (link). That limitation creates a great impact in reading performance. For that reason, during the developer phase of this App, I intentionally limit the maximum number of reading pages to 10. That value can be changed in the "issueController" Class, line 60.

Authentication

In Database you'll notice a table named "Authentication" requesting a valid GitHUb User and Password, that is necessary because the API has a limited number of requests for unauthenticated users (link), also unauthenticated users can't see classified information like Contributors or Stars.

Backend REST

A default REST application, responsible for providing all data needed for the frontend applications.

Libraries

- ➢ Express: Web Framework;
- ➢ Consign: Gives an easy way to organize your Router Files;
- ➢ Axios: An easy to use HTTP client;
- ➢ Sqlite3: Database lib;
- ➢ Crypto-Js: A very complete library for crypto strings and objects;
- ➢ Express-session: Lib to make sessions control;
- ➢ Cookie-parser: A parser lib to read cookies;

Tests and Presentation

In order to facilitate de Debug, tests and presentation of this project, we'll make use of the Insomnia Application (link). A very intuitive and easy to use application, made for debug and tests in RESTful services. The configured Insomnia Workspace file will be available in the documentation folder.