

Computer Science Advising Tool Technical Documentation

Last Place Champions
Rico Adrian, Brian Hooper, Nick Rohde

8th of March 2018

Contents

1	Overview	2
1.1	Problem	2
1.2	Requirements	2
1.3	Design	3
2	Website	4
3	Algorithm	5
4	Databases	6
4.1	MySQL Database	6
4.2	NoSql Databases	6
4.3	Database Handler	6
5	Testing	8

Chapter 1

Overview

1.1 Problem

Design a user friendly website able to pull data from the catalog and allow the user (advisor) to easily manipulate different scenarios for the student based on multiple constraints (i.e. prerequisites, time of the class being offered in the academic year, minimum number of credits, etc.).

1.2 Requirements

1. Login Page
 - (a) Accept credentials from user
 - (b) Verify provided credentials
 - (c) Redirect user to homepage, or display error
2. Advising Page
 - (a) Retrieve existing graduation plan
 - (b) Create new graduation plan
 - (c) Import student transcript
 - (d) Input plan constraints
 - (e) Manually modify graduation plan
 - (f) Print advising documents
3. Administrator Page
 - (a) Modify Databases
 - (b) Create/Modify/Delete users

4. Graduation Plan Generation Algorithm
 - (a) Generate most efficient plan based on constraints
 - (b) Finish within 2 minutes
 - (c) Create a valid plan
5. Databases
 - (a) Store sensitive data in an encrypted environment
 - (b) Store graduation plans for all students
 - (c) Store user credentials

1.3 Design

Chapter 2

Website

Chapter 3

Algorithm

Chapter 4

Databases

4.1 MySQL Database

The MySQL database stores the user credentials, as well as the student graduation plans. Both are stored within the same database, but in separate tables, credentials are stored in the table "user_credentials", and graduation plans in the table "student_plans" (these names can be altered in the "DBH.service" file in the section [TABLES]).

This database is encrypted and cannot be accessed without credentials. All sensitive information is stored within this database.

4.2 NoSql Databases

The NoSql Database (DB4O) stores information about Courses, Catalogs, and Students (only directory information - e.g. Name, Starting Quarter). This database has no encryption, as it **does not** contain protected information, however, it is stored in a binary format making unauthorized access difficult.

4.3 Database Handler

The database handler is the middle man between the databases and the client(s). Its purposes are:

1. **Encapsulate database operations** - It provides query-less database access to all users. Instead, database commands are sent to the Database handler, which translates them into queries. Database commands require no knowledge as to where or how information is stored.
2. **Prevent multiple writes** - two users cannot overwrite eachother's modifications, a user can only update a database entry if it they have the

current version. This is controlled via a "write-protect" property which is part of all database entries. Before a write instruction is executed, the write protect on original and new data are compared, if they do not match, the instruction is not executed to prevent overwriting changes the user may not be aware of.

3. **Manage databases** - The database handler manages the databases for the administrator. The administrator should never need to directly access the databases, except during the creation step. The handler is not capable of creating the Database, however, once the database is created it can create all required tables, and set them up.

Note: The tables have a special format, altering this format by manually editing the database can make the table unusable.

The database handler runs in a multithreaded mode. Specifically:

1. **Master Thread** - A daemon which always runs on the server
2. **Client Thread(s)** - Each client is assigned a dedicated thread which only communicates with its assigned client, and executes all commands for this client. This allows non-critical code to be executed in parallel as much as possible, speeding up operations during periods of high-traffic.
3. **Keep Alive Thread** - A thread which periodically accesses the database to prevent the connection from timing out. This is a result of our testing, as the connection would time out over night.

Upon starting the master thread the password for the MySQL database must be supplied. After the setup has completed, it awaits new clients to connect to it through a specific TCP port (all details pertaining to the connection are specified in the "DBH.ini" file under the [MISC] section). The master thread continues to run even if no clients are connected. Should the service crash, or the server is restarted, the database password must be supplied upon restarting the service to reestablish a connection.

The database handler has two accompanying files, namely, "DBH Setup.pdf" which contains a detailed discussion of how to setup, start, use, and troubleshoot errors pertaining to the Database Handler; and "DBH.ini" which is the configuration file which must be used to start the database handler.

Chapter 5

Testing