Design Documentation

Table of Contents

Problem Definition

“Write a program to organise a collection of code snippets, allowing the user to create, store, delete, view, edit, and search for small pieces of code. Each snippet will be associated with a name, short description and/or set of keywords and a programming language, which can also be modified after creation. The text editing tools should include modifying the text, finding and replacing text and highlighting text.”

Extra Features

- Syntax highlighting
- Extra metadata: Date created
- Sorting of snippets

Overview

The application will be divided into 4 classes. The main class will provide the interface and perform most of the business logic. A second class will interact with the underlying database. The other two classes will act as data structures for snippets and snippet previews. This is helpful as it divides code into two broad categories, business logic classes and data storage classes.

GUI mock-up

The mock up (diagram 1) of what the GUI (Graphical user interface) might look like was draw in the software package ‘LibreOffice Draw’ and helps to visualise

some of the design requirements of the system by showing the actions available to the user.

Database

For the database engine, H2 database engine was decided upon because it is fast, lightweight, can be embedded within a Java application and supports SQL (Structured Query Language). The database will consist of two tables as shown is diagram 2 (EER (Entity Relationship Diagram) Model) and diagram 3 (database schema). The first table will is for snippets and will contain the code for each snippet along with all related metadata. The second table is for tags and will contain the foreign key of the related snippet and the content of the tag. The tag entity is related to the snippet entity by a many-to-one relationship. A database is preferential over storing data in a file as it is more organised, scalable and supports more complex SQL operations that would be difficult or impossible in a text file configuration.

Note to marker: The Software Requirement Specification is 11 pages, however it is possible to fit it into 8 pages without removing any content (by removing white space) but presentation suffers as a result.