Our program is a student information management system. It is designed for the user, representing an administrator, to access and update a database for managing students enrolled at a university.

In the first stage of our project (assignment 1), we laid the basic foundations for our application. After planning and designing our program, we built up the basic infrastructure of the codebase in Java. This included various classes for modelling data, managing the database, managing user interaction with a command-line user interface, and so.

The code functionally enables a variety of uses in our project scope, runs with no known bugs, and follows the SOLID design principles. The app is also user-friendly. However, some design flaws are noted such as poor readability in some places and poor practices such as use of magic numbers. This is attributable to our naivety as programmers at the time. This was our first major programming project. We were unfamiliar with program design and had limited knowledge of object-oriented programming at the conceptual level and of the Java programming language in particular.

In the second stage of our project we implemented a database and a graphics user interface. Lewis Henderson mainly focused on the database and Harry Muir on the GUI but both members contributed to coding collaboratively.

The new version of the program runs with an embedded SQL database implemented with Apache Derby.

The previous versions of our application used a naive approach to modelling and managing the database using file I/O. The tables were encoded in .txt files with each line representing a formatted entry. This approach was overly elaborate with several classes just for parsing the data. On starting up the program, it had to go through a number of elaborate steps to load the database by parsing the text files, validating the data, and reading the data into the data structures in memory. Other classes were needed to update the files by inefficiently deleting and writing lines. It also required a 'double-entry' approach to updating the database, where entries had to be modified in both the text files and in memory in runtime.

All of this was computationally inefficient and was difficult to manage and update, injuring the maintainability of the code base. These problems were swept away by the making use of the Derby database management system. This also provided the advantage of using SQL, a standardised language for relational database management that is more suitable than our previous non-standard, ad-hoc approach.

In addition to this we wrote a graphic user interface. This serves the same basic functions of the UI but in graphical form, providing better usability.

Github

<https://github.com/HazzaGitHubisshit/StudentManageGUI/>