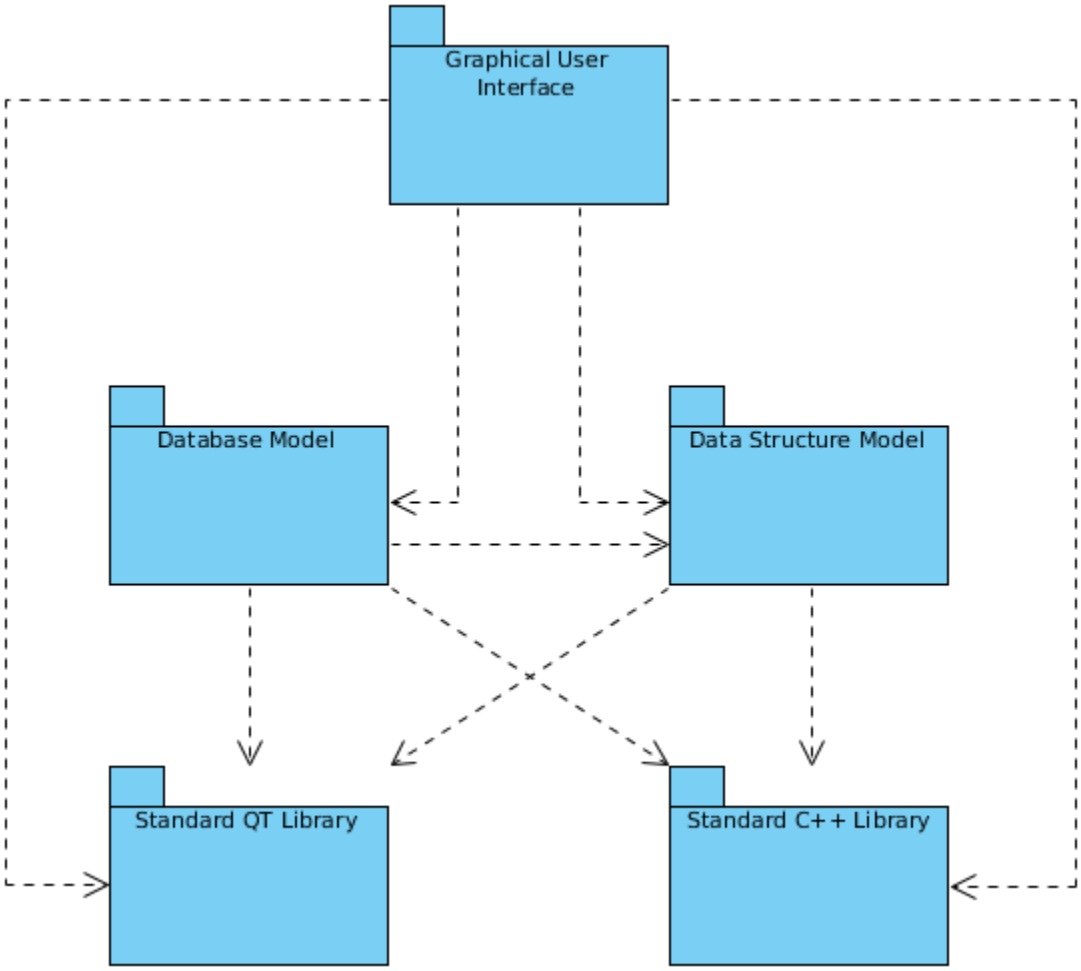
CS3307A – Acuity STAR Project – Team Peach Stage 2 – Package Diagram



Above is the package diagram for the Acuity STAR project application. There is only a single type of relationship in our diagram, the dependency, denoted by a dotted arrow beginning at the source package and ending at the target package and is read "<source package> depends on <target package>". Below is a description of each package and explanation of its dependencies.

Graphical User Interface

This package contains the classes relevant to the the user interface such as MainWindow, PieChartWidget and CustomSort. These classes control the behaviour of all the graphical components (dialog boxes, buttons, widgets, scroll bars, etc.) during user interaction and thus naturally rely on the standard QT and C++ librarys. In addition, MainWindow needs to create new database and data structure models each time the user loads a new CSV file which means it depends on both of these packages.

Database Model

This package contains the CSVReader and RecordsManager classes necessary for the creation and use of the database storing the loaded data. CSVReader is a simple file parser and only relies on the standard C++ library for strings and vectors. However, RecordsManager is more complex as it must create the appropriate database for each dashboard view type (Grants and Clinical Funding, Presentations, Publications and Teaching implemented for Stage 2) in a way which the graphical user interface can easily use. To do this, it is dependant on both the standard QT and C++ libraries as well as the data structure model.

Data Structure Model

This package encompasses the abstract TreeModel class, which makes use of the TreeItem class, and its implementations for each of the implemented dashboard views: GrantFundingTreeModel, PresentationTreeModel, PublicationTreeModel and TeachingTreeModel. TreeModel does make use of RecordsManager when building itself and so the database and database model packages are technically interdependant. Although this goes against the Acyclic Dependency Principle, we belive breaking this rule of thumb is okay as the interdependancy is localized and that, in particular, it does cross the application's layers.

Standard QT and C++ Libraries

These are very well-documented stand-alone packages which are used by, but not in any way dependant on, the other packages in our application.