Ciaran’s primary role was to provide the non-registration servlets (listed below). These servlets provided the Tier-2 interface between the Tier-1 front-end and the Tier-3 database. Their use was ideal due to their ability to open and maintain database connections, rather than reconnecting on each page-load. This reduced the overall connection overheads, reduced database loading, and improved network efficiency. It also centralised the database access, making database issues easier to resolve, and the database itself easier to maintain (for example, when adding additional table entries).

Servlets were therefore written to provide the database access, and straddle the interface between the HTML and the database. They included “business-logic” to provide some of the higher-level database operations (for example, the read-modify-writes in the WithdrawFromServlet) and to map it into the equivalent SQL. This allowed the database access to be abstracted, and private detail to be buried in the Servlets (for example, the database password - this ensured it was not visible to the user).

The list of servlets written by Ciaran, their roles (“Action”) and interfaces (“Input”/”Output”) are listed below.

**DatabaseConnection.java**

Support c*lass to provide the database connectivity of the servlets*

**DeleteUserServlet.java**

*Servlet for deleting users*

**Input** id

**Action** Delete user from the customer table and transaction table

**Output** None

**DepositToServlet.java**

*Servlet for depositing an amount to a user account*

**Input** id, amount

**Action** Read database balance, add amount and update database balance

**Output** Add to transaction table

**Login Servlet.java**

*Servlet for Login Functionality*

**Input** uname, password

**Action** Check the database for the specified username/password and login if applicable

**Output** Writes id, credentials,overdraft and withdrawal limits to session. Calls accountPage/adminPage/login depending on result.

**Logout Servlet.java**

*Servlet for Logout Functionality*

**Input** none

**Action** Invalidates session id and credentials

**Output** Calls login page

**SetMaxWithdrawlServlet.java**

*Servlet for deleting users*

**Inputs** id, maxwithdrawl

**Actions** Sets database maxwithdrawl for user

**Outputs** None

**SetOverdraftServlet.java**

*Servlet for setting the overdraft for a user*

**Inputs** id, overdraft

**Actions** Sets database overdraft for user

**Outputs** None

**WithdrawFromServlet.java**

*Servlet for withdrawing an amount from a user account*

**Input** id, amount

**Action** Read database balance, subtract amount and update database balance

**Output** Add to transaction table