Meadowvale Software Architecture

1. General Software Architecture:

Meadowvale is a Java desktop application. Its front-end GUI is built using the Java Swing GUI libraries and the back-end database is managed by Java Persistent Unit library, as well as the MySQL Java Connecter.

Similar to other desktop software, our meadowvale software is also event driven. When a user triggers a specific event, for instance, pressing a button, a certain event handler in GUI classes (in the meadowvale module) will be called and executed. The function executed usually leads to a creation or a modification of some entity instances (in the entities module), which serve as virtual representations of database tables in Java. If the changes made on entity instances need to be updated in the back-end database, event handlers in GUI classes will call controllers (in the controllers module) to save those changes into the database.

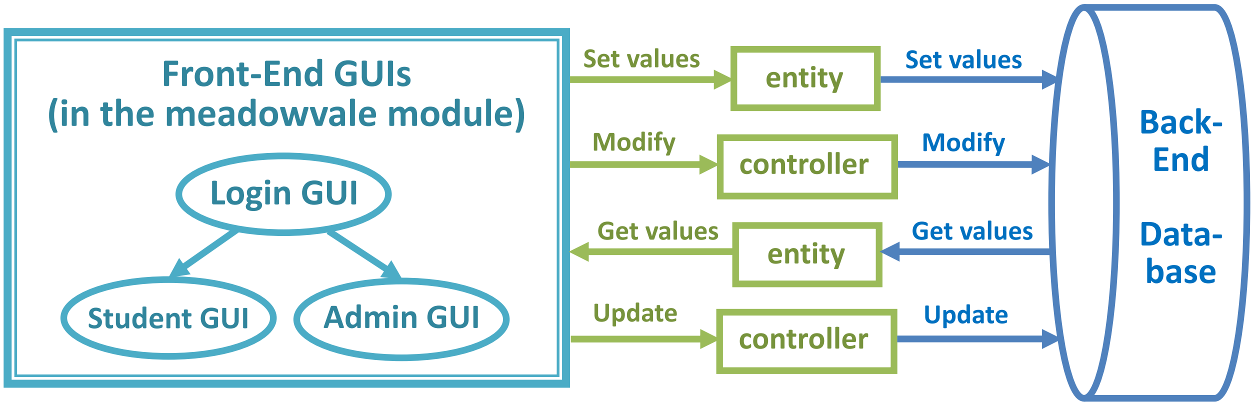
The following explains how we organize our code / software into modules:

(1) meadowvale: This module contains all the front-end GUI classes, as well as all functions that implement software features and logic. It also contains the "application" class, which serves as an interface between the front-end GUIs and back-end database. The "application" class provides GUIs necessary accesses to the database through "entities" and "controllers".

(2) entities: This module contains all the entity classes generated from the database. It creates Java entity objects that serve as virtual representations of database tables in Java. These entity objects let the "meadowvale" module (i.e. the front-end GUI) access and set values in the back-end database.

(3) controllers: This module contains all the "controller" classes corresponding to all "entity" classes. Controllers are used by the "meadowvale" module to easily save the changes made on an entity object (i.e. a virtual representation of database tables in Java) into the back-end database.

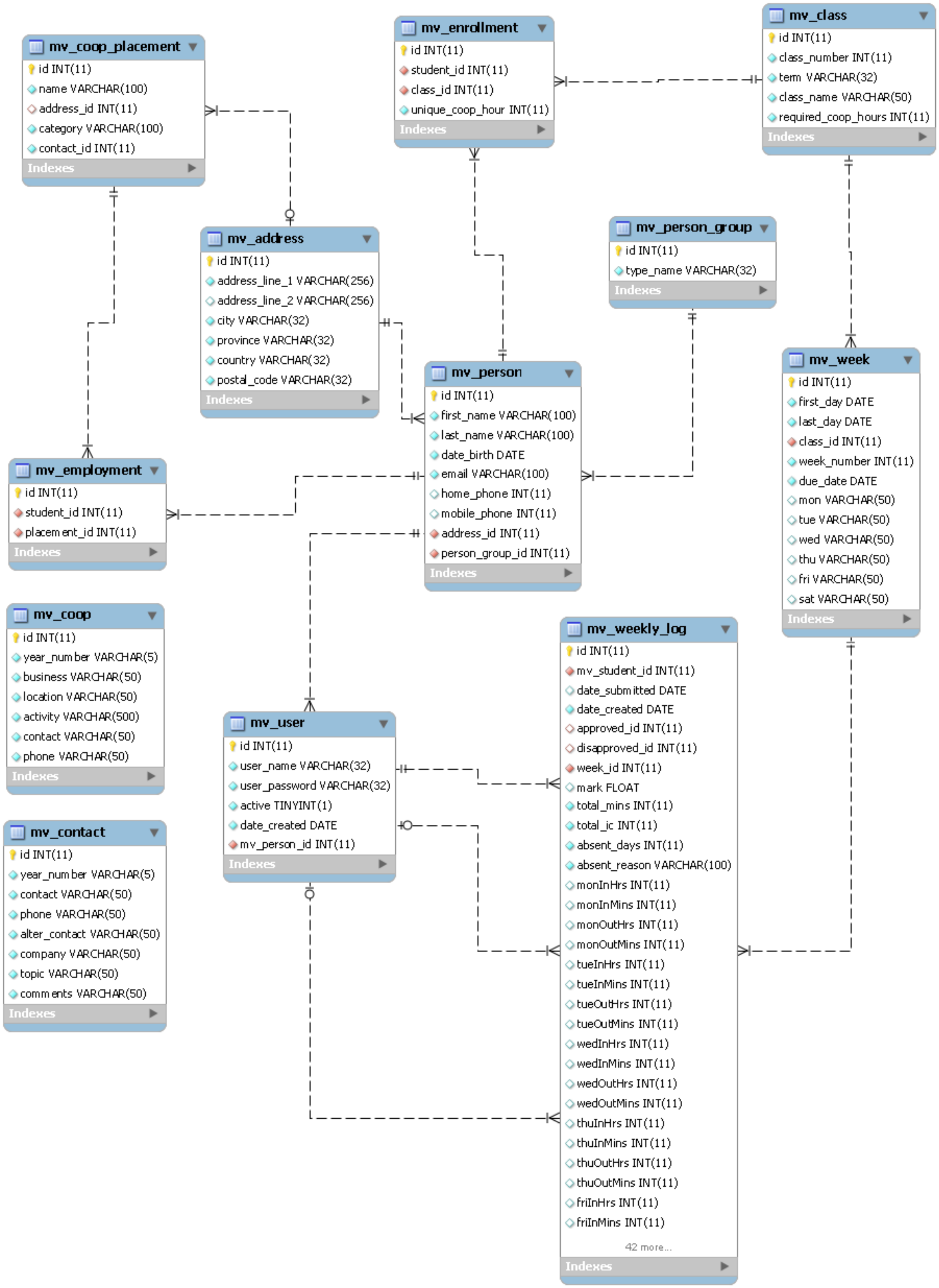
The following diagram illustrates our software architecture pictorially:



2. Database Implementation:

Please refer to the "meadowvale.sql", which we also included in this document directory “Software Related Documents”, for the actual SQL script that is used to construct the database.

The following diagram illustrates our database tables pictorially:



3. External Libraries:

(1) Swing Library: This is used in the front-end GUI classes to create and manage the user interfaces.
(2) MySQL Java Connecter: MySQL API in Java. This is used to connect the software to the MySQL database.
(3) EclipseLink JPA 2.0: Java Persistence API library. This is used in the entity classes and controller classes, which provides GUI classes access to the back-end MySQL database.
(4) Jcalendar: This is used in GUI classes to let the users pick dates on a calendar more intuitively.

4. Existing Bugs / Know Issues:

We have tried also best to fix all the bugs existed in our software. In our final release version, there is one known bug.

In the “Co-op Into” tab of the Admin GUI, it is possible for the user to drag and thus change the order of the columns of the table when filling out and adding co-op job postings information. This could result in error when saving the information in the table into the database as some data will be saved into the incorrect data field.

However, it is reasonable to assume that the user mostly likely will not change the order of columns in the table when filling out the co-op job postings. Furthermore, the potential user of this administrative feature will only be our client and his colleagues. Our temporary solution is simply “Please don’t drag the columns”. We will address this problem in our next release.