PA Framework

A Simplified Approach for Software Solutions

Users Guide

Version 1.0

PA Framework

Architectural Highlights

1. In-memory object model with constant syncing features with the database. Eager loading preffered over lazy loading.
2. All entity relationships handled in real-time at the Business tier rather than the Data tier.
3. Database to store data and data only. All logic that may obstruct database swapping to move to the business tier. Performance enhancement features are OK to be part of the database.

PA Framework Users Guide

- The Framework is implemented as a proof of concept demonstrating the functionality of a representative barebones training institute.
- Framework development is ‘Work in Progress’ expected to evolve and undergo improvements
- Using the Framework application is simple and intuitive.
- Purpose of this guide is to provide help on the usability aspects of the Framework:
  - To provide an overview of the functionality before delving deep into the ‘behind the scene’ customization.
  - To familiarize with how the entities, their child entities, and their parent entities can be browsed and updated with simple create, read, update and delete operations.
  - To familiarize with the built-in out of the box functionality that comes with the PA Framework.
    - History Log
    - User log
    - Customizable Settings
    - Security and Multi-user features
    - Keeping the object model data up-to-date and in-sync with db at all times.

Seven Areas to Familiarize

1. Obtain and Install the Framework application
2. Overview of the User Interface
3. Browse, Add, Update and Delete entities
4. Add and remove child entities from their parent entities.
5. Add and remove parent entities from their child entities.
6. Discover how integrity constraints are handled directly from within the application.
7. Discover the built-in functionality in the Framework.

1. Obtain and Install the Framework application

- Obtain the PA_Institute_Eval_1.0.msi installer from ProgrammingforAll.com.
- Launch Setup.exe from the install folder.
- Keep default selections during setup.
- The application and its Access database will be installed on the Programs files folder ‘PA_Framework’. If needed, it will update the .Net framework to the required version.
- Launch the application to ensure it starts without errors. It will connect to the Access database and load data.

1.1 Launch and modify Settings if needed

- Launch the PA_Institute Demo from Program Files menu and modify Settings if needed
- You will be the sole user for the application, and will have administrative rights to the application.
- Although the demo application is a multi-user application, the evaluation version is single user as all multi-user updating is not fully tested yet.

2. Overview of the User Interface

- The Sample application ‘PA Institute’ contains entities as shown in the data model See Slide R-2. Their relationships are shown in parenthesis along with their associated property.
- An overview of the user interface is shown in Figure R-2.1
- Each entity on the user interface has a separate tab.
- At the left of each tab page, a list contains the main entity associated with the tab.
- On the right side, additional sub-tabs contain lists of child and parent entities for the currently selected main entity.
- Each sub-tab contains two lists, one for the associated entities with the currently selected main entity, while the other lists the available entities which may be added to the currently selected entity.
- All relationships and their types are defined in the meta-data of the solution.
- The content of lists for available child and parent entities is constrained by the type of association they have with the main entity.
- This prohibits users to add any parent and child entity to the main entity that violates the relationship integrity among them.
- Relationship integrity constraint in both directions among all entities is thus established transparently right at the source rather than the database.

3. Browse, Update, Add or Delete Entities

- **Browse Entities**
  - Any entity can be looked up by selecting its tab and browsing through the list on the left. All child and parent associations of the selected entity can be looked up on the sub-tabs for each parent or child association on the right.
- **Updating Entities**
  - Select the entity you want to update, double click on the field you need to update, update the field and move away from it while remaining on the same row. Click on the Update button. A message will be displayed confirming success of update in the database
- **Adding Entities**
  - Click on the Add button, a new entity will be added at the end of the list and also to the daabase. Make updates on whichever fields needing update. While remaining on the same new entity row, click on the Update button. A message confirming the database update will show.
- **Deleting Entities**
  - Select an entity needing to be deleted from the database. The ‘Delete’ button will only be enabled if there is no parent dependency for the selected entity. If required remove all parent entities of the selected entity, to get the Delete button enabled.
  - Click on the ‘Delete’ button. A message will appear confirming the deletion of the entity from the database

4. Add and Remove child and parent entities

- The tab for each entity has sub-tabs on the right for each child and parent entities of the entity.
- Based on the relationships among entities, child and parent entities can be added and removed from any entity. Only valid additions based on the existing relationship integrity constraints is possible on the user interface
- For a 1-1 relationship with a child, only one child or parent can be associated with the entity. Childs or parents that are not already associated with any other entity are shown in the available entities list.
- For a M-1 relationship, only one child can be associated with a parent, while many parents can be associated with a child.
- For a M-M relationship, many child entities can be associated with a parent while many parents can be associated with a child.
- For removing an entity from the associated child or parent list, click on the ‘Remove’ button on the respective ‘Associated Entities’ list. The entity will be removed and added to the ‘available entities’ list below the ‘associated entities’ list.
- For adding an entity to the associated child or parent list, click on the ‘Add’ button on the respective ‘available entities’ list. The available entity will be removed from this list and added to the ‘associated entities’ list above the ‘available entities’ list.

7.1 Built-in Functionality – History Log

History Log

Based on the DB table ‘UpdateLogs’. Every time a change is made to any entity in the database, the frameworks logic adds a row to table ‘UpdateLogs’. This table contains the following columns :

- ID: Key
- DateTime: Date and time of change that took place.
- LoginID: The user who made the change
- TableID: The affected table
- KeyFieldNo: The affected column of the table.
- Operation: Whether the change was N: New, U: Update, D: Delete
- Changes: A comma separated list of fields that got changed, with user definable abbreviations of each field.

7.2 Built in Functionality - Multi-User Features

- Multiple users can access and use the application from different client locations on the network, while the database can be on a central location.
- All database updates are protected against conflicts by the framework.
- Multi-user functionality not fully tested at this time.

7.3 Built in Functionality – User Logs

- Any user logging in and logging out is recorded in the database table ‘UserLogs’. The user ID, login time, logout time and the application version being used are saved.
- This gives the ability to control the application versions installed by users at their client locations.
- This feature will be included in the Framework very soon. Not available at this time.

7.4 Built in Functionality – Customizable App Settings

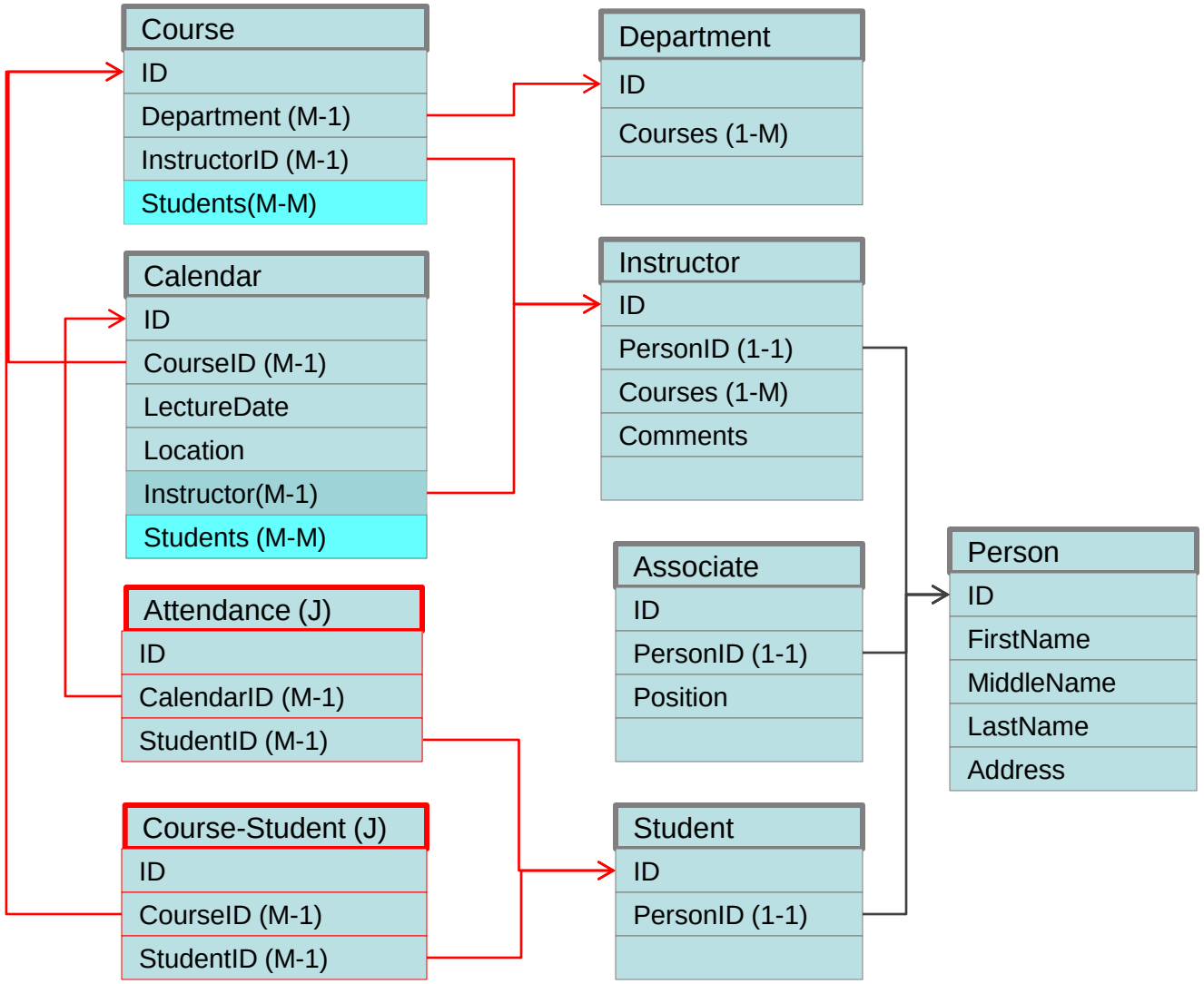
- Settings in the current version are limited, but any setting can be added while doing customization.
  - App settings are stored in an xml file ‘Settings.xml’ in the application folder.
  - Database type and location: Currently Access database is being used, but the application can be switched to any database by changing the connection string
  - Database User ID and password for accessing the database.
  - History Log entries: No of entries in the history log that you want to display in the History screen.

7.5 Synching of Object and Data Models

- Before each update of the database, the local LastUpdateID is checked against the database LastUpdateID.
- If it is the same, the update is done
- If it is different:
  - It means that updates have been made by other users
  - The DB LastUpdateID will be more by the number of individual updates done.
  - Before any update is allowed, the framework brings down all the updates into the local object model.
  - If the current row of the entity being updated is not part of the external updates, the framework will allow the update
  - Else, it will throw a message telling the user to try again.
- The object model is kept in-sync with the database on an intermittent basis (e.g. changing pages or tabs on the user interface.).
  - This is done by comparing the local and database LastUpdateIDs and bringing down all changes (updates, additions and deletions) if any.

REFERENCE

R-2: A Sample Data Model - PA Institute



R-2.1: The Sample app's User Interface

PA Institute UI