

Agave UI Design



[Document History](#)

Version	Date	Author	Description
0.1	1 February 2005	Mike Dameron	Initial version

Table of Contents

OVERVIEW	3
DESIGN	3
<i>General Guidelines</i>	3
<i>Thick-client</i>	3
Guidelines	3
<i>Thin-client</i>	3
Guidelines	3
STORYBOARD	4
<i>Agave core framework startup screen</i>	4
<i>Home panel</i>	5
<i>List Forms</i>	6
<i>Detail Forms</i>	7
<i>Dirty-data validation</i>	9

Overview

The Agave application will consist of “global” navigation and sub-navigation. The global navigation is controlled by the core framework of the application (see Agave Design Spec for more information) and the sub-navigation is controlled within each separate component or functional group application. This document will focus on the UI and navigation within each component.

Design

General Guidelines

- Data entry screens and search ability shall be on separate screens/forms to avoid confusion
- The core framework does not use MDI forms and neither shall the components.
- Accept buttons shall be labeled “OK” and wizard form buttons labeled “Next”.
- On forms that show data and lists, the data shall be shown first and the lists (DataGrids) shown below them.

Thick-client

Guidelines

- All form elements shall be System style to allow for consistent looking controls.
- Every form shall inherit from the associated base form, generated from Visible Developer. The five base forms generated are stdDetailForm, stdListDetailForm, stdListForm, stdMasterDetailForm and stdUIForms.
- Each of the above listed base forms shall use PureComponents NicePanel to provide a consistent and pleasing look and feel.
- Validation shall be handled using the Error Provider control as shown in figure 6.
- Time to load the form shall be displayed in the first panel of the status bar.

Thin-client

Guidelines

- All page elements shall use CSS2 classes to allow for consistent looking controls.
- Validation shall be handled using JavaScript field validation for dirty-data checks and highlight the invalid control in Yellow.
- Time to load the page should be displayed in the footer of the page.

Storyboard

Agave core framework startup screen

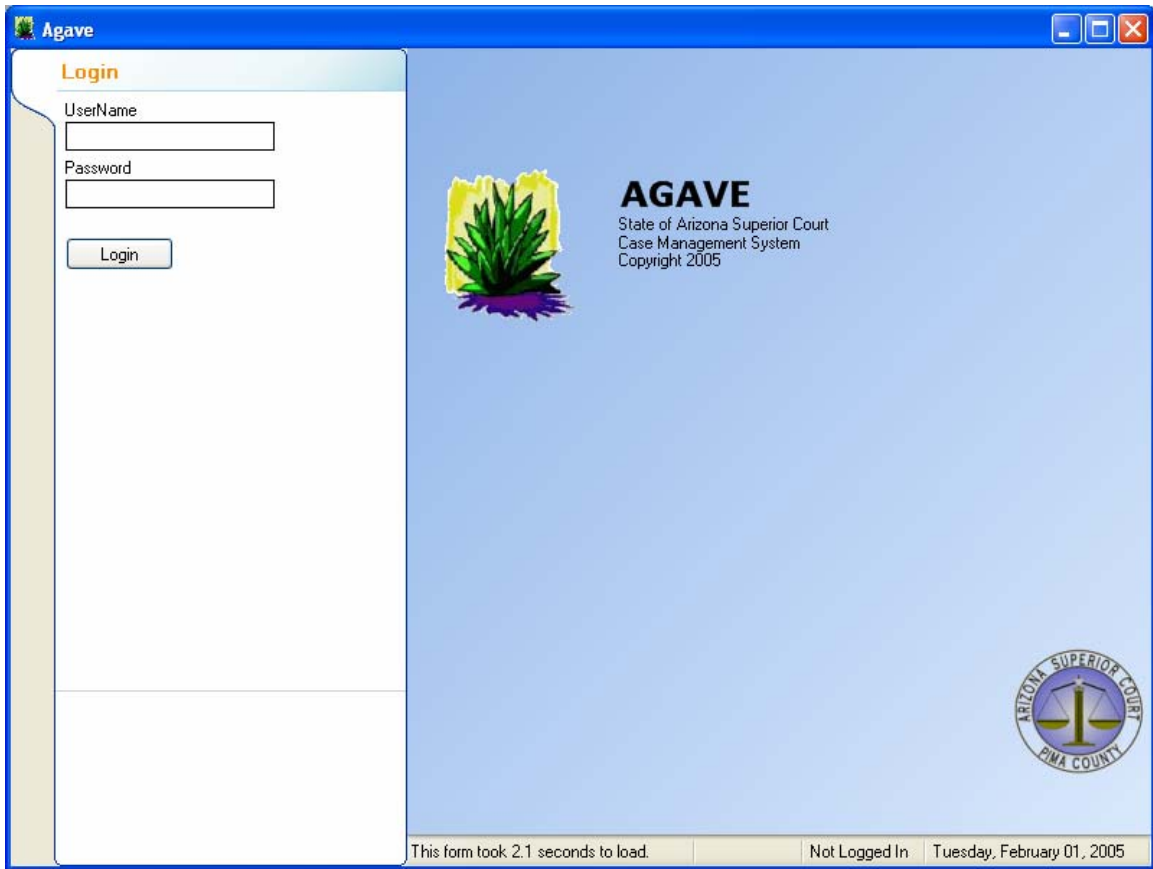


Figure 1 - Splash Screen and Login Form

Home panel

- Navigation panel on left of screen contains only data panels user has access to
- Clicking or double-clicking a panel name launches the associated data panel and hides the navigation panel. Clicking "User Management" generates the screen shown in Figure 3.
- Home panel dashboard widgets, such as "Tasks" shown below contain quick access and provide pertinent information to the user. These widgets may or may not be useful. Some examples would be:
 - Tasks (your current tasks)
 - Calendar (events and tasks pertaining to the users role/group)
 - Favorites (List of most common data panels the user accesses)
 - Statistics (current statistics of the court, which could include # of active cases, # of recently closed cases, # of scheduled events for the day, etc.)

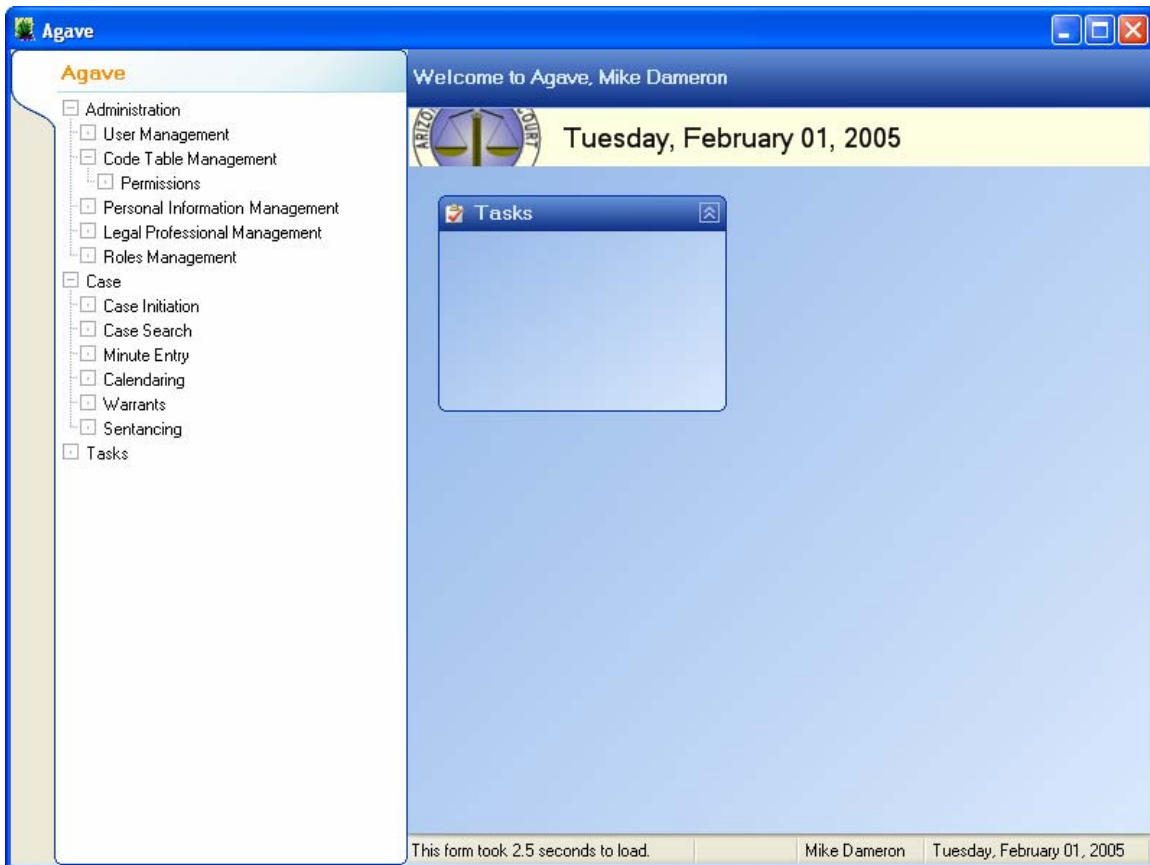


Figure 2 - Home panel, shown after successful login

List Forms

- Form is loaded within the main panel of the Agave core framework
- Toolbar allows user to add, open/edit, search for and delete records
- Header bar contains a bread crumb labeled title
- Header bar contains appropriate icon
- Footer bar shows time to load the form

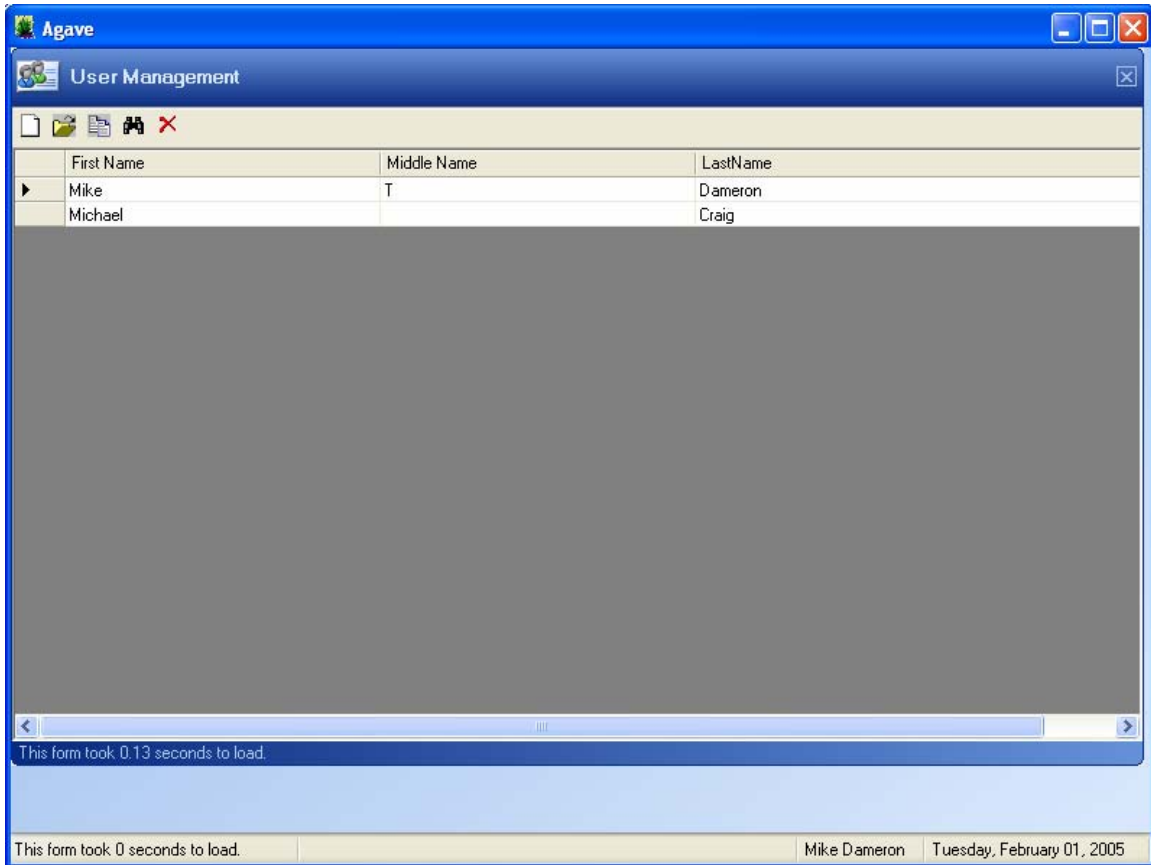


Figure 3 - Administration (FG4), User Management (FG4.1) Data Panel loaded

Detail Forms

- Header bar contains a bread crumb labeled title
- Header bar contains appropriate icon
- Item with focus has background color of yellow
- Data for entity is shown on the top half of the form
- Associated data with the entity is shown in the bottom half of the form with a data grid. If more then one data association exists the data grids will be tab pages placed in a tab control (see Figure 5 below).
- Action buttons for associated data are located at the bottom of the data grid
- Action buttons for the entity data are located at the bottom of the form
- Form has appropriate container image in lower-right corner.
- Footer bar shows time to load the form
- Footer bar shows description of the current control

User Management : User

First Name: Mike

Middle Name: T

LastName: Dameron

Password: [] Password Last Changed: 1/24/2005

Start Date: [] Start End Date: [] End

Force Password Change:

Require Password Change:

User Groups

Group Name	Start Date	EndDate	Default Group
Administration	12/1/2004		Y
Front-Desk Clerk	12/1/2004		

Add Group Edit Group Properties Remove group association

Update Cancel

The person's last name.

Figure 4 - Data Panel detail form with one associated data set

Roles Management : Navigation Group


Group Name: Added by:

Active Code: Modified Date:

Date added: Modified By User:

Data Panels | **Permissions**

Data Panel	DisplayOrder
▶ Case Initiation	1
Case Search	2
Minute Entry	3
Calendaring	4
Warrants	5
Sentencing	6
User Management	7



This form took 0.05 seconds to load.

Figure 5 - Data Panel detail form with multiple associated data sets

Dirty-data validation

- If validation is not valid on the control the `ErrorProvider.SetError` method is called, which displays the red circled exclamation point. The tool tip message of the error provider shown shall be provided and meaningful.
- The control, which failed validation, is colored red.

The screenshot shows a 'User Management : User' form. The 'LastName' field is highlighted in red and has a red exclamation point icon next to it. A tooltip message 'You must enter a last name.' is displayed over the field. The form includes fields for First Name (Mike), Middle Name (T), Password, Start Date, End Date, Force Password Change, and Require Password Change. Below these fields is a 'User Groups' section with a table listing groups: Administration and Front-Desk Clerk. At the bottom are buttons for 'Add Group', 'Edit Group Properties', 'Remove group association', 'Update', and 'Cancel'.

Group Name	Start Date	EndDate	Default Group
Administration	12/1/2004		Y
Front-Desk Clerk	12/1/2004		

Figure 6 – Data Panel detail form showing dirty-data validation