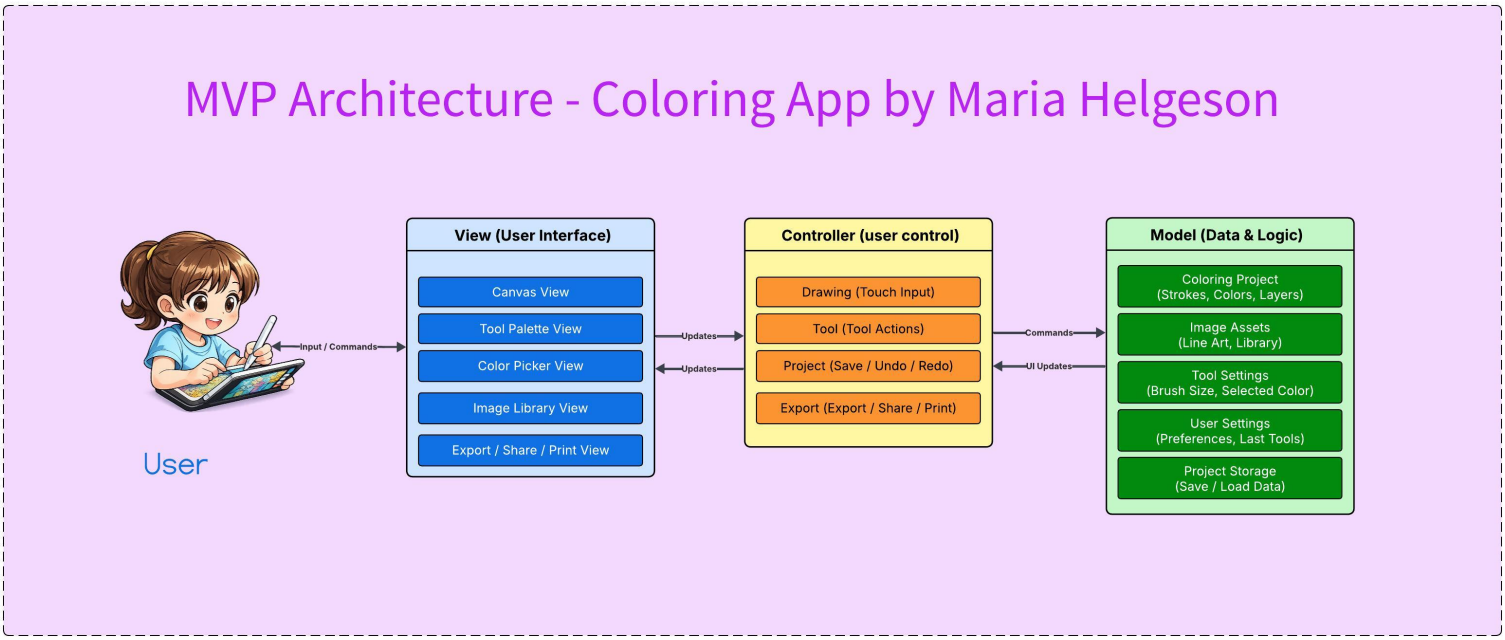
CS3332

Maria Helgeson

Architectural Style: MVC (Model–View–Controller)



I have chosen the MVC architectural style for the iPad coloring application because it provides a clear separation of concerns and supports maintainability as the app grows. Given the application's strong user interface component (drawing, tools, colors) and ongoing user interaction, MVC aligns naturally with how modern mobile apps are designed.

In MVC, the Model manages data and business logic, the View handles the user interface and drawing canvas, and the Controller processes user input and coordinates actions between the View and the Model. This structure makes it easier to update the UI without affecting data logic and to add new features, such as additional tools or export options, with minimal impact on existing code.

Data Handling in the Application

Input data

- User touch input (finger or Apple Pencil strokes)
- Tool selections (brush, pencil, marker, eraser)
- Color selections
- Image selection from the app’s image library

Stored data

- Preloaded line-art images
- User coloring projects (stroke data, colors, layers)
- App settings (last tool used, color palettes)

Processing

- Rendering strokes on the canvas
- Managing undo/redo actions
- Applying selected tools and colors
- Preparing images for export or sharing

Output

- Colored images displayed on screen
- Exported image files (PNG/JPEG)
- Shared images via email or social media
- Print images

Database Requirement

A lightweight local database is required. It would store user projects, image metadata, and settings. This could be implemented using a local mobile database (such as SQLite or Core Data) rather than a full server-based database, since the app primarily operates offline.