PROBLEM 1:

The 2d Image compositor:

COP operations can be easily visualized if we take the scenario of a photo editor, typically Photoshop. The photoshop application would consists of the main drawing canvas, and a set of operations menus. The menu operations consists of various sets of entities like Image selection, Color Palette, Saturation, Hue, Brightness, Lasso/Rectangular Box Selection, 2D Drawing shapes objects, Zoom, Filters, Undo/Redo(COP1). Stacks of Modified frames of the Image/Images, View of the multiple image manipulation Layers(COP2).

When the user clicks on the Image selection/Browse/Open Image button, the image will be loaded on the canvas, the user will have a set of operations to choose from, the user can change the color of the image from color palette option, the user can change the brightness/hue/saturation of the image via the Sliders. The user can also manipulate the images via selecting the parts of the image via a Lasso or selection tool, Zoom in the parts of the images to work on, Apply various filters on the image, Undo and redo its changes and can apply all the COP1 functionalities.

The user can also select/load multiples images on the canvas and can apply COP2 functionalities of manipulating one image over the others, it might be placing/rotating one image over the other, changing the opacity levels of the images, masking the images. The Layers also consists of the above functionalities.

All the operations will be directly reflected on the drawing canvas.

The editor can also keep track of the changes that were being made to the image in the Stack sort of Data structure where the latest modified image will be at the top of the stack with other older modified version of the images down below in the stack.

PROBLEM 2:

BART Train Tracking:

As the user has already visited the BART site via a computer or a mobile device, the user can see the BART dashboard, where the user can track the real time status of the train in the Map canvas which occupies most of the frame in the window to maximize the viewing area and for better visualization.

There are Data tips window box which consists of various train's parameters like its Speed in Miles/hour, current status of the train with the time by which it is delayed/fast or if its on time. The Data tip gets displayed when the user hovers over the image of the train. The user can also see the status with a Green color circle if its on or ahead of time and with a Red color circle if it is running late.

The map canvas has also zoom/panning feature in all the directions where the user wants to select/navigate and zoom into that region/area on the map to better get a view of the current route of the train and/or its stations.

The user can also view the current departing/arriving station, initial station and the final station with the departed/estimated time on the points of the route in the canvas.

The user can also see the dashed virtual image of the train on the same track/route which might be placed ahead/behind the current image of the train depending on whether the train is already running late/delayed or it is ahead of its running schedule. If the virtualized image is on the same image of the train, it infers that the train is running on time.

The train route has already predefined points of its transmission from one point/location(here fremond) to another location(Richmond), this is the case of Data Spotlight, where the user can view the intermediate points of stations on that route shown as the user hovers over the route, so that user can get the feel of when the user can expect the arriving station's time along the user's journey.

On the left side of the canvas, the user can have a set of options to select from. Next stations, Departing stations, arriving stations, depature time, arrival time select dropdowns, upon selecting any of the options, the user can see the real time changes on the canvas, these are the Dynamic Queries which the user can choose from and see the direct manipulation on the map via MVC architecture.