1.

To implement the new railway track I simply created another instance of train and put it on a second track I created, but changed the direction it went in (by editing the move() method and changing the logic for deciding when it went off the screen. As a s result of adding the second train and changing its direction, I had to change the logic in the update() method in CrossingGate, so that it would respond to both trains, and that it would stay closed even if one of the trains told it to open, so long as the other train was still in a position where it had to be closed. I wouldn't really change how I did it.

To implement the cars being able to change roads, I added methods in Simulation.java that dictated how a car would switch roads, by changing to Observers in the original road, removing it from that roads list of cars (which was another addition to the code), and added it to the new road. There was one method to add it from the eastern road to the horizontal road, and then another to add it from the horizontal road to the western road. There were also various different things I had to change in Car.java and associated classes, like adding a way to figure out the direction the car should be moving in. If I would go back and change things, I would change how I implemented the directional stuff. I started implementing the direction of trains with 0 or 1, since the trains would only go east or west, but the cars could go east or south, and I technically could have done it with 0 or 1 again, but doing it that way lessens scalability, and isn't that intuitive, so I should have standardized the directions, and used Direction.EAST, Direction.SOUTH, etc. for both the trains and the cars from the start.

2.

I think most of my design would scale fine, but the major parts that wouldn't scale well would probably be the directions, as I talked about in the previous questions. My directions weren't standardized and could have been implemented better, so that would make it difficult to scale. I also didn't implement ways to move the cars other than in the two needed directions for this assignment, and same goes for the trains. To increase scalability, I should make either one move method that can handle four directions, or four move methods each supporting a different direction, to be called as needed.