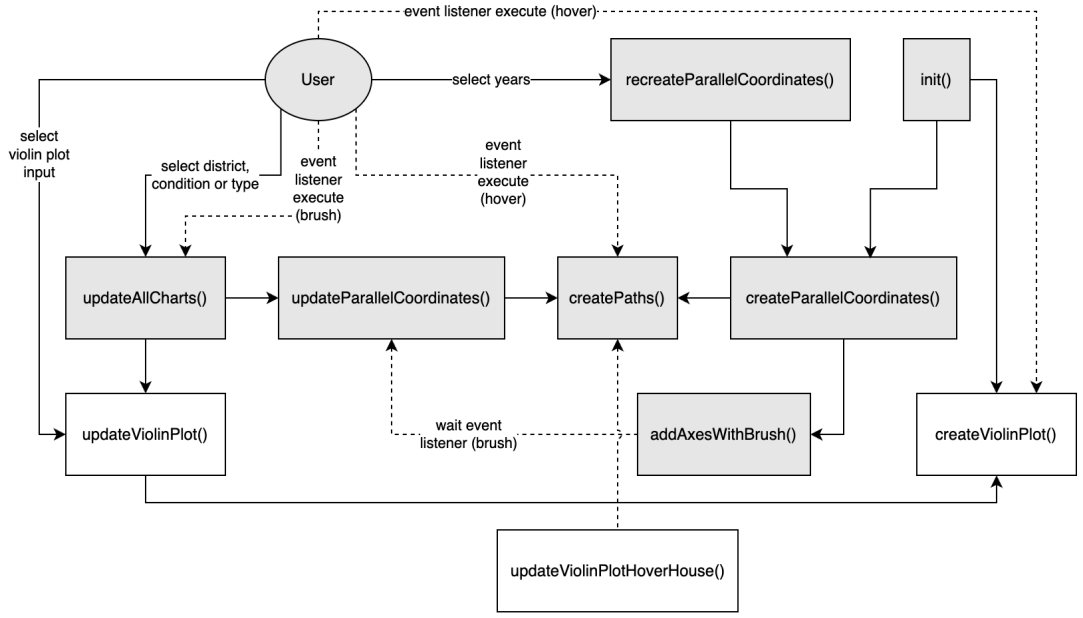
Checkpoint IV: Second Prototype

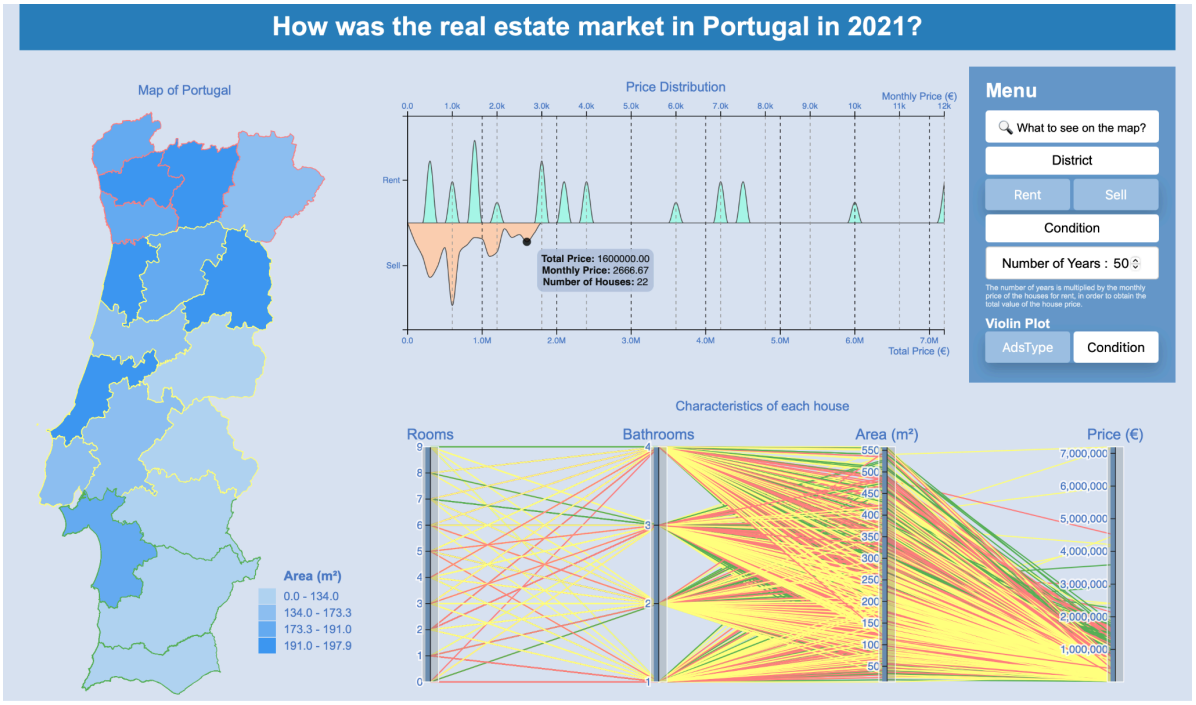
Group: G23
Date: 2024/10/18

Prototype Architecture (the image is changed)



Regarding the last checkpoint (gray boxes), the methods **createViolinPlot**, **updateViolinPlotHoverHouse** and **updateViolinPlot** have been added to the file **violin_plot.js**, since this file is responsible for implementing the functionalities of the Violin Plot.

Dashboard Layout (the image is changed)



Regarding the dashboard layout presented in the previous checkpoint, the differences include the implementation of the Violin Plot (replacing its old placeholder), the specific explanation of how rent prices are calculated, and the addition of buttons in the controllers menu to allow interaction with the Violin Plot.

Regarding the previous checkpoint, we also decided to add the specification of the unit of measurement (m² for area and € for price) and a small border around the values in the parallel coordinates so that they are always visible.

Data Processing

For this checkpoint, we only made a small change to the data processing explained in the previous checkpoint: removing outliers from the prices. We noticed that our function to remove price outliers was not working correctly, as it was only removing outliers from house prices for sale and not from rental prices.

Chart Interaction

This chart interacts with two buttons on the controller, allowing to compare the **types of ads** (AdsType: “Rent” or “Sell”) and the **conditions of the properties** (Conditions: “New” or “Renovated”). When clicking on one of the buttons, the selectViolinPlot(show) function is triggered, where the `show` parameter indicates the chosen comparison.

This function calls the updateViolinPlot(data, selector, show) method, which recreates the violin plot according to the option selected. In this way, data analysis becomes dynamic, allowing the user to explore different categories simply and efficiently.

We have added a tooltip interaction for each point on the graph, displaying specific information about the monthly price, annual price, and the number of houses available. This interaction is triggered by the **.on(mouseover)** function, which detects when the user hovers over a point and displays the corresponding tooltip with the relevant details (view in the dashboard layout).

Compared to the previous checkpoints, we've also added a tooltip for Parallel Coordinates to represent the respective information.

Chart Integration

We maintain both the global and filtered data in **script.js**, but since users may want to view only houses for sale in the Parallel Coordinates and Choropleth Map, the Violin Plot data cannot be filtered by AdsType or Condition when those dimensions are selected. To address this, we introduced a new variable, **violin_data**, which filters information based on these restrictions. This variable is updated along with the filtered data and is used by the Violin Plot similarly to how the filtered data is used by the Parallel Coordinates.

For the integration of Parallel Coordinates with the Violin Plot, we added an interaction where, when the user hovers over a path in the Parallel Coordinates, a corresponding circle appears on the Violin Plot, indicating where that house is located. This interaction is handled by the **updateViolinPlotHoverHouse()** function. Whenever the Parallel Coordinates detects a **.on(mouseover)** event on a path, it triggers this method to display the circle on the Violin Plot, highlighting the selected house.