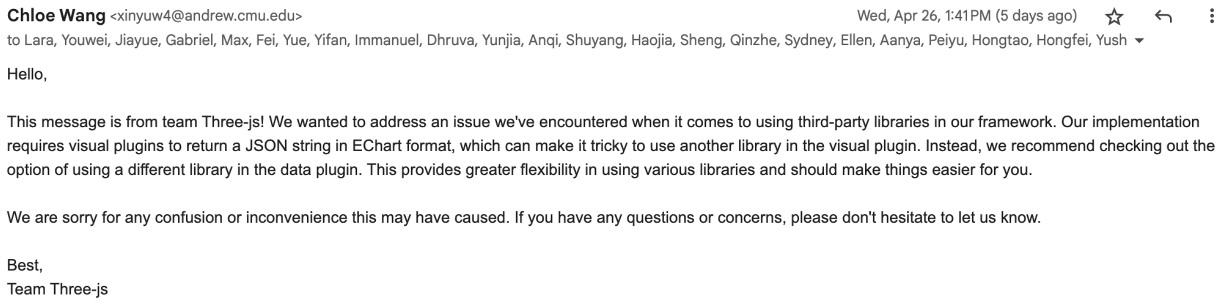
Before releasing the framework, we made significant improvements to our ReadMe.md file, adding more detailed and specific information on all the classes likely to be used by the plugins. Additionally, we removed information about API tokens and keys, and instead provided detailed instructions on how to generate and use them within the framework's code. We also provided instructions on how to load plugins by referencing the specific file path that plugin teams need to edit. We think because our documentation was rather comprehensive, and we have comments for every function in our implementation, there weren't a lot of questions asked regarding the framework implementation.

Although our experience providing support for the plugin teams was in general smooth, the primary concern we had in the process was the design limitations in our visual plugin interface. Specifically, we had restricted the visual plugin to return a JSON string in the EChart library format, which hindered plugin teams from utilizing other third-party libraries in the visual plugin. Recognizing this limitation and its potential conflict with the assignment requirements, we sent an email to all students implementing our framework, recommending them to use a third-party library in the data plugins instead. The email we sent:



Chloe Wang <xinyuw4@andrew.cmu.edu> Wed, Apr 26, 1:41PM (5 days ago)

to Lara, Youwei, Jiayue, Gabriel, Max, Fei, Yue, Yifan, Immanuel, Dhruva, Yunjia, Anqi, Shuyang, Haojia, Sheng, Qinzhe, Sydney, Ellen, Aanya, Peiyu, Hongtao, Hongfei, Yush

Hello,

This message is from team Three-js! We wanted to address an issue we've encountered when it comes to using third-party libraries in our framework. Our implementation requires visual plugins to return a JSON string in EChart format, which can make it tricky to use another library in the visual plugin. Instead, we recommend checking out the option of using a different library in the data plugin. This provides greater flexibility in using various libraries and should make things easier for you.

We are sorry for any confusion or inconvenience this may have caused. If you have any questions or concerns, please don't hesitate to let us know.

Best,
Team Three-js

Additionally, a student on Github raised an issue regarding dead code in our implementation of App.js that we had forgotten to comment out. We addressed this by informing the student that those functions were not used by the plugins. This issue taught us the importance of adhering to good code style even for files that are not to be edited by plugin teams, to avoid any potential confusion.