Maintenance Plan

As with any type of software product, software maintenance is one of the most important factors to think about, as it can determine whether the product will be successful in the long run or not. Thinking about the maintenance for the first year after the initial deployment would be the first step. The cost of this maintenance could vary greatly, depending on the growth of the user base of the product, but nevertheless all types of maintenance should probably be necessary. After the initial release, even though the software was in a good and functional version, perfective maintenance should be performed. This means that the software should be further enhanced and improved to have an even better performance than the on it had at release time. This could be based upon our personal perspectives on what should be added or what should be improved or modified from the initial version, or also based upon user feedback on certain aspects of our software that could be even better. This could come in the form of adding more possible ingredients to the application, increasing the amount of filters, etc. or perhaps redesigning some aesthetic aspects of the web interface or modifying the layout of different buttons. During the first year after deployment, perfective maintenance would probably be the one that is performed the most by the team, in order to accommodate to user feedback and to have the most enhanced version possible. With the perfective maintenance the performance should be improved and therefore the size of the client base should also get bigger. That's why most of the maintenance cost would come in the form of the perfective maintenance. On the other hand, with the user feedback some bugs and faults in the program, which were not detected by our tests, could be discovered. Since a good amount of testing was performed before the deployment, the amount of faults should probably not be too big, but since no amount of testing can certify that a software is bug free it is likely that some faults would in fact be discovered. This means that corrective maintenance should also be needed during the first year of our app. The amount of corrective maintenance would depend on the amount of fault found in the code, so the cost of this maintenance could vary widely, but it would probably be the second biggest cost of maintenance. Finally, adaptive maintenance could potentially also be needed during the first year of the product. Since our application runs with an API call, our product depends on this particular API, so in the case that this API where to suffer changes or modifications our team would need to adapt our product accordingly. This would probably be the only scenario where adaptive maintenance would be needed, therefore it would account for a smaller portion of the maintenance cost. Since our product doesn't require any fees in order to operate, as it is a web app, there would not be any costs related to servers or distribution platforms. Given the amount of maintenance needed for our project, possibly more developers could be needed, so hiring one or two new team members would also be part of the maintenance cost. Therefore, the overall cost of maintenance of our product during the first year should be composed of the salary of the new developers hired, plus the cost of performing the needed perfective, corrective and possibly adaptive maintenance. All these costs should be covered accordingly to ensure a successful cycle for our product.