**MVC Scientific Calculator**

A Project Improvement Proposal

in Information Technology 3

Presented to the Faculty of the

College of Computer Studies

Lyceum of Alabang

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Information Technology

By:

Cresencio, Rodolfo Miguel C.

De Luna, Mart Dexter S.

Gerolia, John Gabriel S.A.

Sarmiento, Jerico J Q.

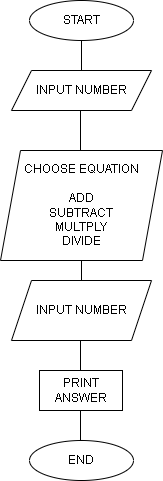
Tabin, Jasper P.

MARCH 2024

**Introduction**

The previous project that we made last prelim was a Simple MVC Calculator. This upcoming midterm, we plan to improve this project into something more complex. The MVC Calculator that we made has only basic mathematic equations and we want to add more advanced functions.

**Current Process Analysis**



The current process in this project is just simple calculator. You can see in the flowchart the process is just entering a number then choose if you want to add, subtract, multiply, or divide. Then you choose another number to have a total of the number you input.

By improving the calculator, we can add more functions to the calculator that it would make it as a scientific calculator.

**Proposed Improvement**

* **Enhanced UI:** To create a smooth and easy navigation through the calculator. Adding new functions to the current UI of the calculator to easily navigate the functions.
* **More Advanced Functions:** Add new functions on the calculator like unit conversion and scientific functions.

**Benefits & Impact**

The Model-View-Controller (MVC) design is useful for scientific calculator applications because of its separation of responsibilities, modularity, improved user experience, and collaborative development. It encourages a more ordered codebase, reusable components, and the smooth integration of new features. MVC also makes unit testing, debugging, modification, and collaboration easier, making it an excellent choice for developing powerful calculator applications.

**Project Plan & Timeline**

If ever this topic gets approved, then these are the plan and timeline we the team are going to do.

**Timeline:** 4 weeks, setting an iteration in GitHub with a span of 4 weeks.

1. **Breaking down the project into pieces:** Assigning a group member to an issue and set it to the iteration to “Midterm Project” which consists of 4 weeks of development.

**2.** **Reviewing the Issue:** Once the member asks to review the pull request, it will be reviewed and check if there are any bugs or mistakes.

**Equipment needed:** Computer/Laptop.

**Evaluation & Risk Assessment**

There are various aspects to consider while evaluating a Model-View-Controller (MVC) scientific calculator. These include feature evaluation, user interface evaluation, performance evaluation, error handling and validation, and quality assurance, as well as documentation. By methodically evaluating these factors, you may detect potential flaws, prioritize mitigation techniques, and ensure the overall quality and dependability of your MVC scientific calculator application.

**Conclusion**

The Model-View-Controller (MVC) design gives benefits such as modularity, enhanced user experience, scalability, and collaboration. However, it involves assessing standards and dangers. These include functionality, usability, performance, error management, security, quality assurance, and maintenance. Calculation errors, user experience concerns, performance obstacles, safety concerns, and maintenance are all potential risks. The MVC scientific calculator's success depends greatly on continuous improvement.