Group 3: Jahlil Owens, Trishelle Leal, and Landon Strappazon

Written By Trishelle Leal

Dr. Ruth Lamprecht

CMSCI 349 A

September 20, 2024

Milestone 1b

**Deployment Engineer**

As a deployment engineer, I have set up an initial repository on Git to store all group documentation and source code. Folder and branch structure has been initially set, but as the project grows, it will change or add branches or folders. To be able to have contributions from different members of the group, we have opted to use GitHub. The repository is private, to ensure that nobody but the group members and the professor have access to said repository.

Merging code can be difficult, so we have opted to create a process to be able to ensure the code that is committed is correct and to facilitate the identification of possible issues within the project. We have decided that our branching strategy will work using the following methods:

* The main branch will only be used for stable, production-ready code.
* Those working on an individual feature will create separate branches for their work. This is to recognize files per feature and the features that are still not ready to be deployed.
* The develop branch will function as a pre-stage for the main branch, where integrating features can be tested prior to merging.
* The developer will submit a pull request to be able to merge their code into the develop branch. Each pull request will require a code review and approval.

As the person in charge of deployment, I will leverage the merging between branches, making sure that the group members’ code is correct and able to work with what is already integrated into the branches. On another note, I will manage the Integration environment. Once code is merged into the develop branch, it will be deployed to the integration environment for testing. All team members will be set as responsible for ensuring that their code works within this environment before requesting a merge onto the main branch.

**Git capabilities**

Capabilities are essential for a collaborative environment, and to ensure code integrity, avoid conflict and promote the creation of good, efficient code.

* **Branching:**  The group members will work within isolated branches to avoid interfering with each other’s work. This allows for parallel development and editing without creating any conflicts.
* **Push:** Each will push their code to their feature branches and open pull requests when needed. This ensures that only reviewed and approved code is integrated.
* **Merge:** This is a capability only I will have. After code reviews, I will merge into develop and later onto main.

PyCharm’s built-in Git functionality will be integrated with GitHub for easy-version control. Using this IDE, the group members can easily use Git capabilities within the interface. Additionally, PyCharm’s diff and merge tools can be used when resolving conflicts or reviewing changes in the code. This will ensure that the project that is delivered towards the end of the semester is of great quality and that the group members can create it seamlessly, without conflict.