Branching Strategies

Group Three - Case II

I. Branching in Version Control

I.I About developing your booking systems:

In the following paragraphs we will explain how we can profit from observing the idea of branching in our group work.

I.II About SOFTWARE development:

- What is a branch?
  A branch is essentially a unique set of code changes with a unique name. Each repository can have one or more branches. The main branch — the one where all changes eventually get merged back into, is called **master**. This is the official working version of your project and the one you see when you visit the project repository at *github.com/yourname/projectname*.
  When you create a new branch, you reference back to the branch you created it through. This means if we choose to create a new branch from the *master* one, we would reference back to the *master* branch, and create a copy of the project that tracks new changes which might happen from the original reference.

- Do you see possible usefulness in your current project work?
  Absolutely! It makes it easier to work together on the same project.

- What is a 'pull request'?
  A **pull request** is a request of getting your feature branch pulled into the master branch (or any other branch) you're making a *pull request* for. A *pull request* is a merging of two branches that requires review from another person. PRs are frequently used in larger open source projects online, when people want to contribute to the code base, owners of the code (or similarly authorised individuals) review the code and either accept the *pull request* or reject it with comments of what is not good enough.

- What is an 'issue'?
  Issues can be used to track bugs, ideas and enhancements in projects. They can as well be used as task management, and provide an interface to discuss topics uncertainties between team members.

- What is a 'blame'?
  Blame in *git* can be used to point out who made certain changes to location (lines) in the code.

- Do those concepts find use in your development cycle?

Yes, as of now, we've made use of *branching*, *pull requests* and *issues* within our team.

I.III About GROUP WORK dynamics:

- What is a branch?
  A branch is a way to separate your work from the other people in the group. It makes you able to work isolated on a feature, so concurrent development can be present. However, multiple people working on the same branch can cause problems if good communication is not established - people might start to change the same files, and this might result in complicated merge conflicts.
- Do you see possible usefulness in your current project work?
  Yes! Branching in our project would help us with:
  - Not be afraid of a branch breaking;
  - Track new changes in each branch and make it easier to rollback or debug;
  - Create a team strategy for an easier implementation of new features (Continuous Integration or Continuous Delivery).

- What is a ‘pull request’?
  A **pull request** in terms of group work, allows the other people in the group to have a look at your changes before they get merged. It allows for proper peer review to happen, to learn from each other, and to make sure that code goes through the review process that might very well cause fewer errors in the long run.

- What is an ‘issue’?
  An **issue** for group work is a thought, idea, bug or whatever that you'd like to point out to the rest of the team. Usually, *issues* can mean a lot of things and it is the team’s responsibility to determine how they are going to use these in the team.

- What is a ‘blame’?
  When a lot of people work on the same project, it's nice to know who made certain changes, so you know who might have introduced a bug, or deserves a "thumbs up" for the great implementation. **Blame** makes it possible to see and inspect the code difference. This makes it easier to track and also understand the changes made by the author.

- Do those concepts find use in your development cycle?
  As mentioned earlier, we have used almost all of them, except for the *blame feature*. Considering that the software development hasn’t really begun yet, we haven't really developed much individually on the project. We've only worked all together or in smaller groups as of now, in order to establish the contract between the frontend and backend systems. We expect to continue developing our practices of these concepts as they will be useful in our project - we would make a decision in the near future about how to handle our work environment further.