Developers Guide

BarcQ collaborators

December 2020

1 Introduction

This guide is written for developers to understand the development workflow, and software architectures

2 Git workflow

Whoever want to contribute to BARCQ, needs to have a github account. Github allow all the contributors to work on the same project, this will reduces the code conflict. The development workflow is as follows:
With your username clone the BarcQ repository

```
git clone http://github.com/rsehgal/barcQ
```

when you clone the github repository, you will get a local repository in which you can work.

After cloning one should create its own branch with some logical name.

```
git checkout -b branchName
```

It will be nice if one follow some naming convention like including github username and issue number in your branch. For example a user with name **test** working on **issue-XX** may create a branch

```
git checkout -b test/issue-XX
```

This command will create a local branch for the user in which one can implement the feature on which he/she is work. Its always a good practice to keep on committing your changes and pushing it to the repository, so that in future if you want than you can go to any version of the code. Code can be committed using following commit command

```
git commit
```

This will open a light weight editor where you can write the commit message. The commit messages are very important, so please write the commit message that should reflect the work that you have done. Till now the code is committed in your locate repository. In order to reflect new code on the web, one should push the code using followig command

```
git push origin branchName
```

Now enter your credential to push to code.

If the developer thinks that the feature, which he/she is working on is complete then he/she may create the pull request, that signifies that you want to merge your changes to master branch. Few people can be made responsible to do the final merge. Currently since the repo is public hence any collaborator can merge the code to master, but this merging work needs to done by 1-2 people only, who before merging the code should actually the review the code and test it for the feature introduced by developer. If everything is OK then only the code should be merged to the master
(Warning : Developers, please just create the pull request but don't merge it yourself to the master branch, let somebody review the code and then merge)

3 Selecting an issue to work on

On the github repo, there is an **issues** tab, where one can see the issues that needs to be worked on. So a developer can chosen anyone and before start working on it, assign it to yourself, so that nobody else should work on that.
If somebody found some issues with the web portal, then please create an issue on the github repo, and write few lines of description, and it will be nice if one also writes the procedure to reproduce the issue.