**Branching workflow for GITLAB**

**1)master**- This branch is the origin of all branches containing the production code. All development code is merged into master sometime. Parallel to master another branch is develop

**2)develop-** This branch contains development code.when the source code in the develop branch reaches a stable point and is ready to be released, will create a release branch

**3)release**\*- Release branches support preparation of a new production release.This branch can be used for testing phase also for code deployed with **QA/UAT/PERF** team. May branch off from develop and must merge into master and develop.

**4)feature-** Feature branches are used to develop a new feature for the upcoming release. When the features are finished then they are merged into the develop.

**5)hotfix\*-** hotfix branches are necessary to act immediately upon an undesired status of release branch after deploying to the production. May branch off from release and must merge into master and develop.

**Naming convention for Release Branch and Hotfix Branch**

* Release Branch name will be as Release-<Year>.<Month>

**Ex:- R-20.2**(R-Release, 20-Year,2-Month)

* Hotfix Branch will be as Hotfix-<Year>.<Month>

**Ex:- H-20.2**(H-Hotfix, 20-Year, 2-Month)

**Folder Structure for Branches**

**1)Platform-** This folder contains the code related to Platform.

* Terraform Scripts, Dockerfiles,CICD files,etc.,

**1.1) API**

**2)Entity\_Extraction-**

**1)Extraction Doc-type**

**1.1)1040**

**1.2)W2**

**2)PostProcessing(Optional)**

**3) Classification-**

1. **Training-**
2. **Dataset Creation**
3. **Model Training**

**2.1-Retraining**

**2) Inference-**

1. **Model Inference**
2. **Subdocument Grouping**
3. **Intelligent Document Splitting**

**4)QA-**