|  |  |  |
| --- | --- | --- |
| Use Case | FeatureService | 1.0 |
| Description | This Service hosts all the necessary methods and properties managing the feature(s). | |
| Pre-requisite | The project must exists. | |
|  | Create a service with the following methods:  **GetFeatures(projectid) –** This method should gets all the active feature(s) from the feature table by the projectid. Use the projectid as input parameter.  **GetFeature**(featureid) – This method should get the feature by the id.  **AddFeature(Feature) –** This method should add a new feature. Use the “feature” table. Set the “active\_ind” in the feature table as 1 (true) and the key using the UUID.  **ManageFeature(Feature)** - This method should update the feature changes. Use the “feature” table.  **RemoveFeature** – This method should delete the feature. There should be any task associated to the featureid. | |
| DAO | Create the DAO as per the standard. | |
| Feature table | CREATE TABLE IF NOT EXISTS `feature` (  `feature\_id` INT NOT NULL ,  `project\_id` INT(11) NOT NULL COMMENT 'This is features table\n' ,  `name` VARCHAR(100) NOT NULL ,  `description` VARCHAR(500) NULL ,  `key` VARCHAR(36) NOT NULL ,  `priority\_code` VARCHAR(45) NOT NULL ,  `active\_ind` VARCHAR(45) NOT NULL DEFAULT '1' ,  `due\_date` DATETIME NULL ,  `date\_added` DATETIME NOT NULL ,  `date\_modified` DATETIME NULL ,  `created\_by` INT NULL ,  `updated\_by` INT NULL ,  PRIMARY KEY (`feature\_id`) ,  INDEX `feature\_Project1` (`project\_id` ASC) ,  CONSTRAINT `feature\_Project1`  FOREIGN KEY (`project\_id` )  REFERENCES `Project` (`project\_id` )  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB | |
|  |  | |