Software Requirements Specification

Version 1.0

FeatureMesh

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of the Feature Mesh System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## 1.2. Scope of Project

This software system will be a Cloud native feature store . This system will be designed to maximize the Data Scientists productivity by providing a platform to assist in feature engineering where the push and the pull of features from the platform is provided as a service along with other value-added functionality described later in the document. By maximizing the data scientist’s and Data Engineer’s work efficiency and production the system will meet their needs while remaining easy to understand and use.

# 2.0. Requirements Specification

This section outlines the use cases for each of the users separately.

### 2.2.1 Data Scientist Use Cases

#### Use case: Publish Feature (Core)

**Brief Description**

The Data Scientist accesses the platform and publishes the built feature on the platform.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Data Scientist has already developed the required features in his development environment and is ready to push the same on the platform. Also, the required Library and/or use GUI is built in to the platform for Data Scientists to use.

* The user specifies metadata for the features that would be used while searching.
* The user leverages the GUI or the library to push the developed feature onto the platform.
* The system displays the status message of the push operation to the user.

#### Use case: Search Feature

**Brief Description**

The Data Scientist accesses the platform and searches for the intended feature from the platform.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Data Scientist has already published the feature to the backing store on the platform.

* The user chooses to search by feature name, keyword, timestamp etc.
* The system displays the choices to the user.
* The user selects the desired feature from the search results.
* The system presents the abstract of the feature to the user.
* The user chooses to re-use the feature.
* The system provides the requested feature.

#### Use case: Offline Store

**Brief Description**

The Data Scientist accesses the platform and uses the published feature for model training using the offline store within the platform.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Data Scientist has already developed the required features and published the features on the platform.

1. The user leverages the built library from featuremesh to get the values of a feature for training the analytical model in his environment (i.e: Jupyter-Notebook etc)
2. The query parameters for the library call would include the feature Name and any other associated meta-data of the feature.
3. The system responds back with either a success status with associated feature value or an Error
4. The user has to program to catch any exceptions/errors thrown by the platform.
5. The workload for Offline store typically would involve Batch processing.

### 2.2.2 Data Engineer

#### Use case: Online Store

The Online store is a separate module on the platform that facilitates real time feature consumption for inferencing done by the analytical model deployed in production.

**Initial Step-By-Step Description**

Before this use case is initiated, the data scientist has already provided his go ahead while publishing the feature to the online store as well.

<< More to be added w.r.t how a model in production would refer the online feature store>>

## 2.1 Requirements

The platform on which the above-mentioned services would be built on is Kubernetes. The online store is required to be extremely fast with low latency for any of the feature value access coming in from the inferencing model. The platform must leverage service mesh for Authentication and authorization constructs and other security aspects like auditing, RBAC etc. Basic logs emitted from the core service mentioned above should be easily pluggable into Observability tools available on Kubernetes platform.