# Vision

## Project Proposition

Quality knowledge capturing is an objective set by the AMS team. Part of this objective requires capturing high quality information contained in Jira issue description and resolution comments. Successfully capturing this information should assist in increased knowledge retention, faster member integration, and improved organizational agility.

## Problem

Jira issues do not always contain “high quality” information. Determining which Jira issue information is high quality and which are not is unrealistically time-consuming for someone to do manually. A system needs to be put into place to automatically evaluate Jira issues and provide a rating of high or low quality.

## Proposed Solution

The proposed solution is the Knowledge Enhancement and Evaluation Project, otherwise known as KEEP. KEEP will be comprised of two main pieces, KEEP App and KEEP Training. KEEP App will depend on the text classification model trained by KEEP Training, but otherwise both pieces of KEEP can be installed and operated in silo from one another.

### KEEP App

A REST API and Jira custom plugin to assist in the collection and evaluation of Jira issue information. The features include:

REST API

* Provides an API endpoint
* Feeds input to a text classification model
* Returns prediction results

Jira Custom Plugin

* Automatically detect issue creation/updates
* Collect Jira issue description and resolution data
* Forward data to REST API
* Accept and display returned prediction results

KEEP App will use the following hardware and software:

* Web server running Microsoft IIS for REST API hosting
* A running and configured Jira instance
* Python and Java as the principle programming languages
* Pytorch and Fastai as neural network frameworks
* Django as a REST API framework

### KEEP Training

A neural network training application to assist in the process of obtaining training data and training text classification models. The features include:

* Collect Jira issue data from a Jira database
* Preprocess Jira data for model training
* Create a language model
* Fine-tune a text classification model
* Export a trained text classification model

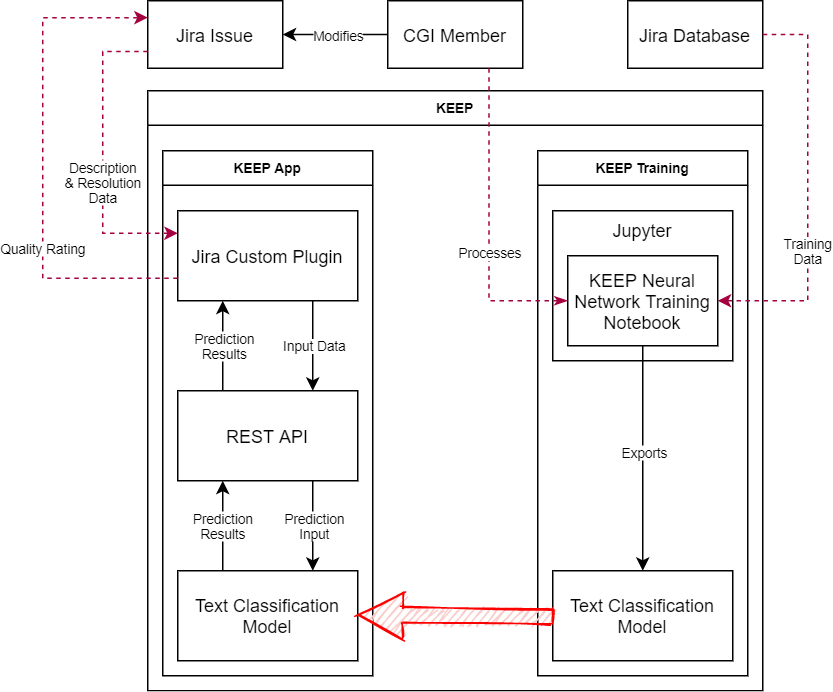
KEEP Training will use the following hardware and software:

* Data server with an accessible Jira database
* Computer running Jupyter for Jupyter notebook processing
* Jupyter as an interface for neural network training
* Pytorch and Fastai as neural network frameworks

## Actors

|  |  |
| --- | --- |
| **Actor** | **Goals** |
| CGI Member | * Evaluate the quality level of Jira issue description and resolution information |

# System Diagram

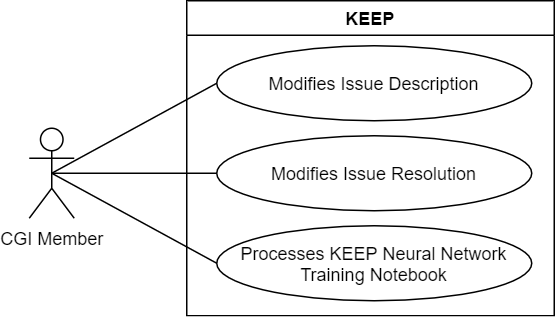


# Use Case Model

## General Actor Descriptions

|  |  |
| --- | --- |
| **Actor** | **Description** |
| CGI Member | **Role**: Any CGI employee. Supplies Jira issue description and resolution information, or retrains the KEEP text classification model.  **Responsibility**: A CGI Member interacts with the system for the purpose of obtaining a quality rating on the description and resolution information on a Jira issue, as well as, processing the KEEP neural network training notebook. |

## Diagram of Use Case Model



## Use Cases

### Modifies Issue Description

#### Brief Description

This use case describes a CGI Member obtaining a quality rating for a Jira issue after creating, or modifying the Jira issue description.

#### Use Case Narrative:

A CGI Member wants to obtain a quality rating for a Jira issue after changing the description field. The CGI Member provides the following:

* Description

When completed, the CGI Member submits the Jira issue update request to the system. The system processes the request and forwards the description to the KEEP text classification model for prediction. When completed, the system returns a quality rating to the CGI Member.

#### Special Requirements

* N/A

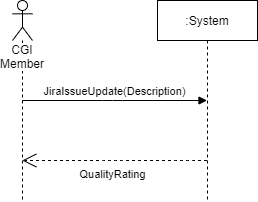
#### Preconditions

* CGI Member has access to the system
* KEEP text classification model is accessible

#### Post Conditions

* Quality rating is returns to the CGI Member

#### System Sequence Diagram



### Modifies Issue Resolution

#### Brief Description

This use case describes a CGI Member obtaining a quality rating for a Jira issue after creating, or modifying the Jira issue resolution.

### Processes KEEP Neural Network Training Notebook

#### Brief Description

The purpose of this use case is to allow a CGI Member to process the KEEP Neural Network Training Notebook in Jupyter, thereby training the KEEP text classification model.

# Risk List

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Name** | **Priority** | **Description** | **Mitigation Strategy** |
| Poor Dataset | Critical | Data contains lots of useless, misleading, or unrelated information and requires heavy amounts of pre-processing. | Dataset will need to be parsed and analyzed to remove unneeded content, formatting and punctuation. |
| Issue Definition | Critical | There is no clear definition of what makes an issue “high quality”. | Project leadership will need to define what makes an issue “high quality”. |
| Flawed Model | Critical | Due to the Poor Dataset and Issue Definition risks, the current model is verified to not be working as intended. Currently the model always reports a low quality rating for any issue. | Poor Dataset and Issue Definition risks will need to be addressed first. The model will need to be re-trained and tested using a cleaner dataset. |
| False Positives | High | Neural network models are prone to producing results that look correct, but are in fact false. This is especially true with NLP. | Careful testing and verification will be required once a candidate model is produced. A verification period after initial implementation is recommended. |
| Inexperience | High | Alec is inexperienced with neural networks and training custom models. Alec is inexperienced with the complex mathematics required for deeper customization or understanding of model architecture. | Careful research and experimentation to learn new skills. |

# Software Architecture Description

## Use Case View

### Architecturally Significant Use Cases

* Modifies Issue Description
* Modifies Issue Resolution
* Processes KEEP Neural Network Training Notebook

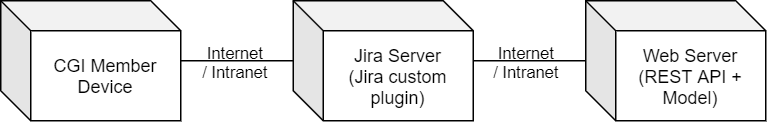
### Discussion and Motivation

These three use cases represent the main methods of interacting with the system and are therefore considered to be architecturally significant. The main complexity of the system comes not from the use cases, but from the nature of understanding and training neural network models.

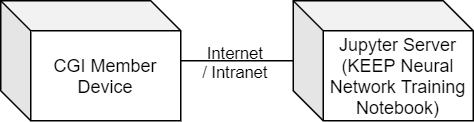
## Deployment View

### Architecturally Significant Nodes and Physical Configuration between Nodes

**KEEP App**



**KEEP Training**



### Discussion and Motivation

KEEP App and KEEP Training are physically separate sub-systems. Once the KEEP text classification model is trained using KEEP Training, it will be manually copied to KEEP App for use. Therefore, each sub-system can be represented separately in the deployment view.

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AMS Team |  |
| API | Application programming interface. Defines the interaction points (endpoints) between multiple software systems. |
| Django | A Python-based free and open-source web framework. |
| Endpoint | An interaction point on an API, generally in the form of a web URL or address. |
| Fastai | A simplified neural network framework running on top of Pytorch. |
| IIS | Internet Information Services. A web server software created by Microsoft. |
| Java | Class-based, object-oriented programming language. |
| Jira | Issue tracking product developed by Atlassian for project management. |
| Jira Custom Plugin | A custom code library extending the default operation of Jira. |
| Jira Issue | Jira’s representation of a unit of work. |
| Jupyter | An open-source web-based application that allows the creation of documents, called notebooks that contain live code and narrative text. |
| KEEP | Knowledge Enhancement and Evaluation Project |
| KEEP App | The REST API portion of KEEP |
| KEEP Training | The portion of KEEP responsible for training the neural network models. |
| Language Model | A machine learning model trained to predict the probability of a sequence of words. |
| NLP | Natural Language Processing. The study of interactions between computers and human language. |
| Python | An interpreted, high-level and general-purpose programming language. |
| Pytorch | Open-source machine learning library developed mainly by Facebook’s AI Research Lab. |
| Quality Rating | A rating (high or low) assigned to Jira issues by KEEP. |
| REST | Representational State Transfer. A software architecture that defines constraints for creating web services. |
| Test Classification Model | A model trained to categorize text into specific groupings of tags or categories. |
| Training Data | The initial set of data used to train a machine learning model. |

# Test Cases?