Title: Extracting Requirements Specification from Code (ReqEx)

Members:

- Zachary Bruggen zbruggen2016@my.fit.edu
- Nicholas Epler nepler2018@my.fit.edu
- Ivan Hernandez ihernandez2018@my.fit.edu
- Thomas Morrison tmorrison2017@my.fit.edu

Faculty advisor:

Dr. Slhoub kslhoub@fit.edu

Client:

Dr. Slhoub

Date(s) of Meeting(s) with the Client for Developing this Plan:

- Wednesday 1/19
- Formal meetings with advisor established on a biweekly basis

Goal and motivation:

In the Software Development Industry and in Academia, Developers don't have an easy way of understanding the requirements of a system without numerous hours testing and manually evaluating the system. To make this process more efficient, our goal is to create a system that can generate system requirements by analyzing each component of the software. This will allow Developers to gain a much better understanding of any open-source software or code that was not developed by them in a much quicker fashion.

Approach (key features of the system)

Easily comprehend code functionality

- Speed up time understanding code functions by not having to manually analyze code to get close to the original requirements. The analysis will be displayed to the users in an easy to understand format.

Display Key functionalities and properties of the source code

- Displays additional information besides the generated requirements that were used in the making. This allows Developers to continue enhancing their understanding of the code base by only focusing what was deemed necessary.

Text Analysis for source code files

- Generates lexicological analysis for each word in the source code file. This information can be used by Developers to understand and identify text patterns in the source code files.

Able to input one or multiple files of source code

- The system will be able to parse and manipulate basic language patterns from multiple files at once. If comments are present in the files, they will be used for the analysis as supplemental data. If only one file is detected, the system will proceed and develop its own requirements using any information found in the source.

Algorithms and Tools:

- Freeling text analyzer for Text Analysis
- U++ for GUI development
- Dijkstra's Algorithm for shortest path in the original code between sections.
- API that calls from the code, in order to link to the database and send over the parsed information as it comes out of the original code that is being parsed. This API is hosted by Github and utilizes the key that Github provides.

Novel features/functionalities:

This code will be designed in a way that can help reverse engineer system requirements that are within the code, understanding the syntax and returning an easily readable comprehension for the code. This is helpful in industries as to identify previously completed work and how it operates. Potentially saving clients money and time in the software development cycle.

Technical Challenges:

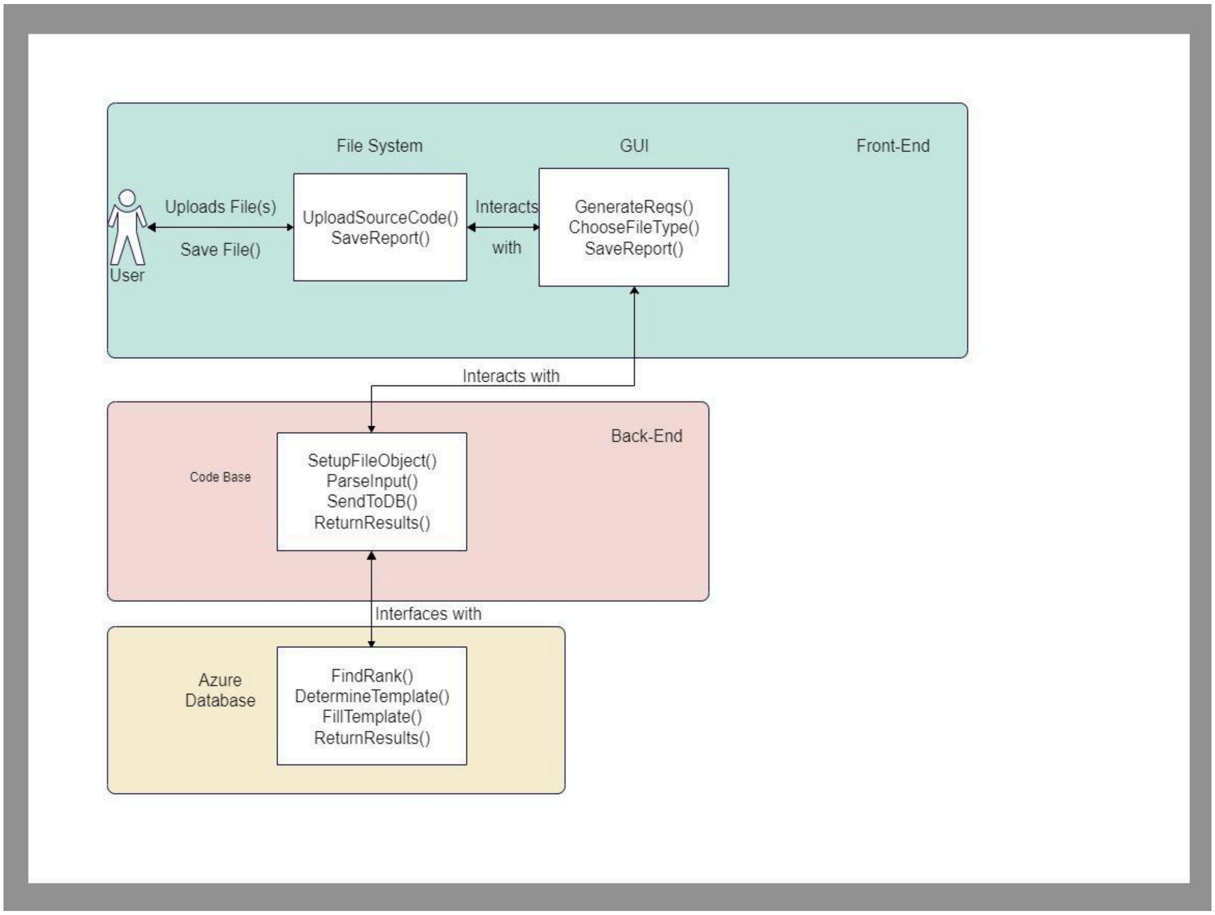
A key discovery that we had during the first semester is that as we consider more complex input cases, we need to implement more functionalities. A technical challenge for this semester will come from adding these functionalities without compromising the speed or accuracy of the system.

This system is going to be using some sort of GUI to present easier interaction for the user. Not many of us have experience in GUI development, so this will be a challenge for us to overcome.

A major technical challenge of the system is the organization and structure of the database. All of the information that is being parsed from the code is being sent to the database, in which it is being organized. As of now the database is able to organize the information, it just needs now to be tested for accuracy in the system so it is able to operate as it should.

We talked earlier in the planning phases about how we wanted to extend this system to multiple languages. Currently, we are only focusing on C++, so depending on how quickly we can complete the system, a challenge will arise in applying this system to multiple programming language conventions.

Design: System Architecture Diagram



Evaluation: How to Measure Success?

Because our system is going to be generating requirements from an automatic analysis, a good measure for the system's success is the reliability and accuracy of the system. In particular, we are going to generate requirements like that of a user would create, to see how reliable the system is and scale the accuracy. As well, a correct measurement of the success of our system is the efficiency, being able to have the code, parse the data, and return the information in a timely manner is important to the overall system.

Progress Summary:

Module/feature	Completion %	To do
Storage	90%	Add/Update functionality as it appears
Text Analysis	90%	Add/Update functionality as it appears
Database	75%	Organize the information and test for accuracy
Research	50%	Research is still in progress

Milestone 4 (Feb 14): Itemized Tasks:

- Establish communication with all of the components of the project
- Generate a prototype requirement
- Begin developing the GUI and displaying key information

Milestone 5 (Mar 21): Itemized Tasks:

- Finish GUI
- Update requirement generator(Database) to produce more accurate results
- Evaluation results
- Create poster for Senior Design Showcase

Milestone 6 (Apr 18): Itemized Tasks:

- Update and test requirement generator(Database) for result accuracy

- Test/demo of the entire system
- Evaluation results
- Create user/developer manual
- Create demo video

Task matrix for Milestone 4 (teams with more than one person)

Task	Zach	Nick	Ivan	Tom
1. Research	25%	25%	25%	25%
2. GUI Development	40%	20%	40%	0%
3. Update/Configure Code Base	50%	0%	50%	0%
4. Update/Configure Database	0%	0%	0%	100%
5. Software Testing	10%	70%	10%	10%

Description (at least a few sentences) of each planned task for Milestone 4:

Task 1: Research

Our client requested that we perform research into similar projects, in order to determine if our system is unique. We will be testing different projects we research, and determine their similarities to our own project. If we develop a unique project, it is in our client's interest to create a paper detailing our system for possible publication.

Task 2: GUI Development

Our goal for Milestone 4 is to generate a simple requirement, so we want to have a basic GUI to display that information. The plan is to have a basic GUI to allow for user input, and to display the generated requirement. In the later Milestones we will then update the visuals of this GUI to provide more functionalities, and an overall better user experience.

Task 3: Update/Configure Code Base

As we consider different inputs, we need to implement additional functionalities to reach those cases. This task applies to adding these features into the code base, in order to handle different complexities in the input

Task 4: Update/Configure Database

To match the code, the information that is being sent over to the database needs to be able to parse lots of data at once, as well as structure and organize the data. As of now the information is ranked but not tested for accuracy, the information now needs to be tested for accuracy and create finite changes to match the accuracy.

Approval from Faculty Advisor

- "I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."
- Signature: _______________-________________________ Date: _________________