Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Document Overview 4

1.1 Purpose 4

1.2 Document Conventions 4

1.3 Team Contact information 4

1.4 References 4

2. Project Mission Statement 4

2.1 Project Introduction 4

2.2 Product Vision and Scope 4

2.3 Stakeholders 4

2.4 Assumptions and Constraints 4

2.5 Business Requirements 4

3. Requirements 4

3.1 Functionality 4

3.1.1 Provide individualized help and feedback 4

3.1.2 Intelligently pick the best questions for the student 4

3.1.3 Analyze Student's Code 4

3.1.4 Provide Understandable Errors 5

3.2 Non-functional Requirements 5

3.3 Performance Requirements 5

3.4 Security Requirements 5

3.5 Software Quality Attributes 5

3.6 Supplementary Requirements 5

4. Classification of Functional Requirements 5

Provide individualized help and feedback 5

Intelligently pick the best questions for the student 5

Analyze Student's Code 5

Provide Understandable Errors 5

5. Use Cases 6

6. Sequence Diagram 7

7. Appendices 7

Appendix A: Project Glossary 7

# Document Overview

## Purpose

The purpose of this document is to detail the requirements and specifications, in order for the client to better understand the project. This document contains a list of all information relevant to the project.

## Document Conventions

None

## Team Contact information

## References

# Project Mission Statement

## Project Introduction

The purpose of this project is to build an Intelligent Tutoring System to help college student learn the introductory concepts of computer programming. The majority of Computer Science programs have a low retention rate after the first course of in program. The Intelligent Tutoring System will use Python to teach students, which will provide them hands on practice solving a variety problems. Also the system will be able to provide individualized help and feedback for each student.

## Product Vision and Scope

The Python Tutor is aimed at aiding students taking their Introductory to Programming course. This project is not only limited to college students but anyone who wants to learn the concepts of programming.

## Stakeholders

## Assumptions and Constraints

## Business Requirements

# Requirements

## Functionality

### Provide individualized help and feedback

The Tutoring System will be able to provide individualized feedback and help based on the student's learning styles. The feedback and hints will be provided in a variety of ways and based on the students response to the system will tailors it's responses most beneficial to the student.

### Intelligently pick the best questions for the student

The Tutoring System will be able to intelligently the next best question for the student. The System will gather information about the student, such as if the student needed help, how close they were to the correct answer, and how long it took them. Base on the information the System will be able to pick the best question to ask the student next.

### Analyze Student's Code

The Python Tutor will parse through and analyze the student's to ensure that sound programming principles are followed. The system will contain a few possible ways to code each problem and will compare the student's code those stored in the system. This is important because the output maybe correct, but the student may have not used sound programming principles. For example, if the student is to calculate x raised to the third using the built in math functions, then they could write pow(x, 3) or pow(x, 2) \* x. Both of these two statements achieve the same the same thing, but one is better than the other.

### Provide Understandable Errors

The Tutoring System will present to the students their error, if any, in a much more readable and easier to understood way than the regular errors produced by the interpreter. Often the errors that the interpreter provides are not easy to understand and can cause confusion and frustration to the student. The system will contain some common errors for each type of problem and provide an error message that best explains the problem with their code.

## Non-functional Requirements

## Performance Requirements

## Security Requirements

Students login information must be keep secure and confidential.

## Software Quality Attributes

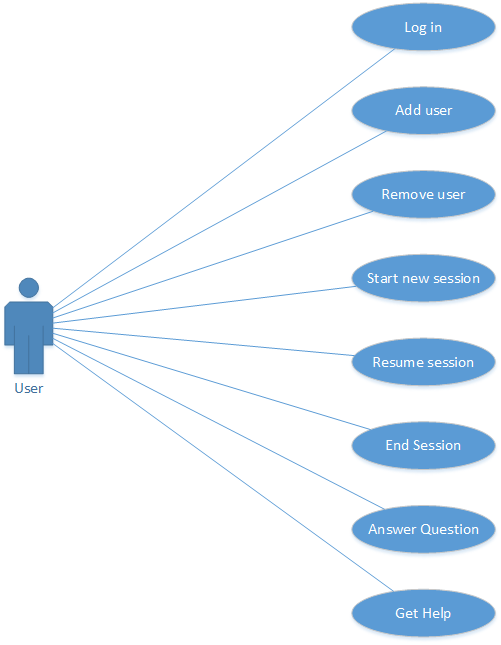
## Supplementary Requirements

None

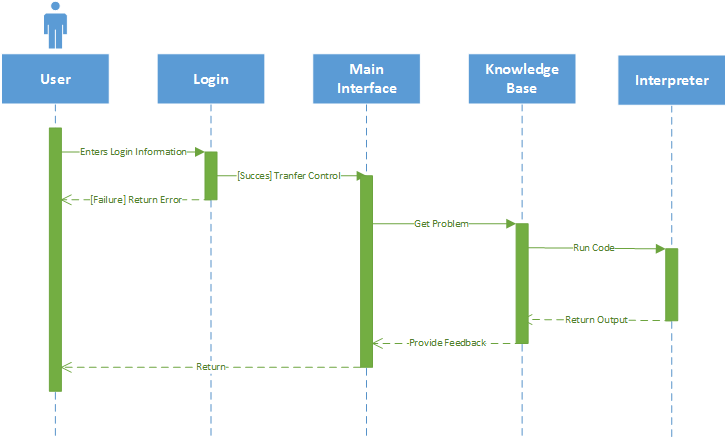
# Classification of Functional Requirements

|  |  |
| --- | --- |
| **Functional Requirement** | **Priority** |
| Provide individualized help and feedback | Essential |
| Intelligently pick the best questions for the student | Essential |
| Analyze Student's Code | Essential |
| Provide Understandable Errors | Desirable |

# Use Cases



# Sequence Diagram



# Appendices

# Appendix A: Project Glossary