CODE CRITIQUER FOR C

TEAM INFORMATION

- Project ID: sdmay24-34
- Team: Nicholas Carber, Conner Cook, Brandon Ford, Emily Huisinga, Sage Matt, Cade Robison
- Advisor: Dr. Rover
- Client: Dr. Ureel, Michigan Tech University

PROBLEM/NEED

- Compiler messages are daunting
- C compilers alone do not provide helpful feedback
- Provide a helpful experience for fixing antipatterns

CONTEXT

- Our goal was to help novice programmers learn C by creating an application for students to receive feedback on their code
- We also wanted to allow instructors to customize feedback to be tailored to their own assignments
- The application is hosted on an accessible website

DEFINITIONS

- Antipattern - Incorrect code or poor style decision
- Regular Expression - Pattern that detects strings
- XPath - Query language for XML documents
- AST - Representation of the structure of code

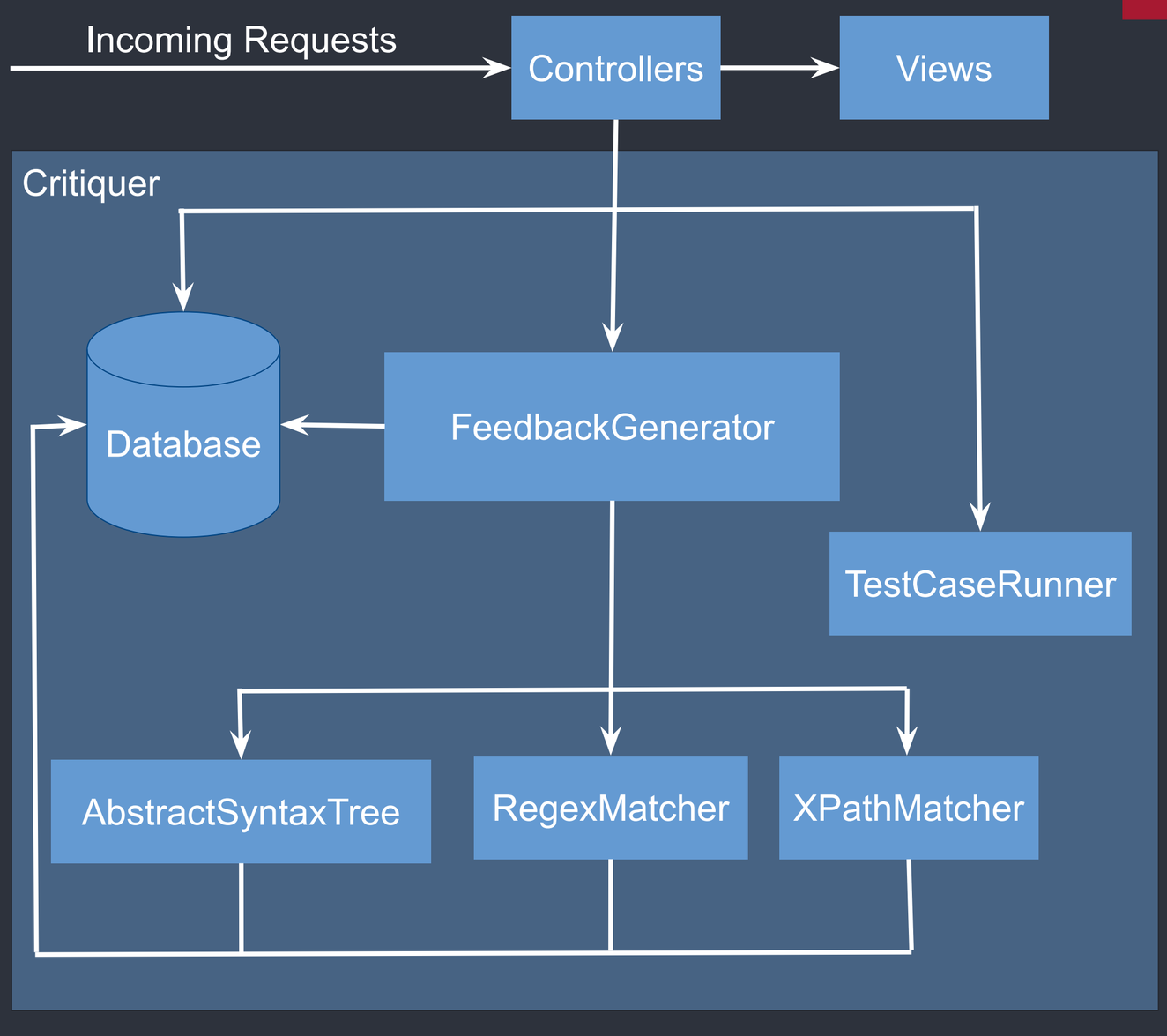
REQUIREMENTS

- Implement both regular expressions and XPath antipatterns
- Support individual assignment creation and access
- Allow instructors to upload test file that can be run against a student's code

DESIGN & TESTING

- Students interact with web-application
  - Upload code
  - Receive feedback
- Instructors have tools to fine tune the application
  - Create assignments
  - Create custom antipatterns expressions
  - Select which antipatterns to use
  - Upload custom tests
- Testing
  - Unit tests for each function
  - Manual acceptance testing for front-end

COMPONENT DIAGRAM



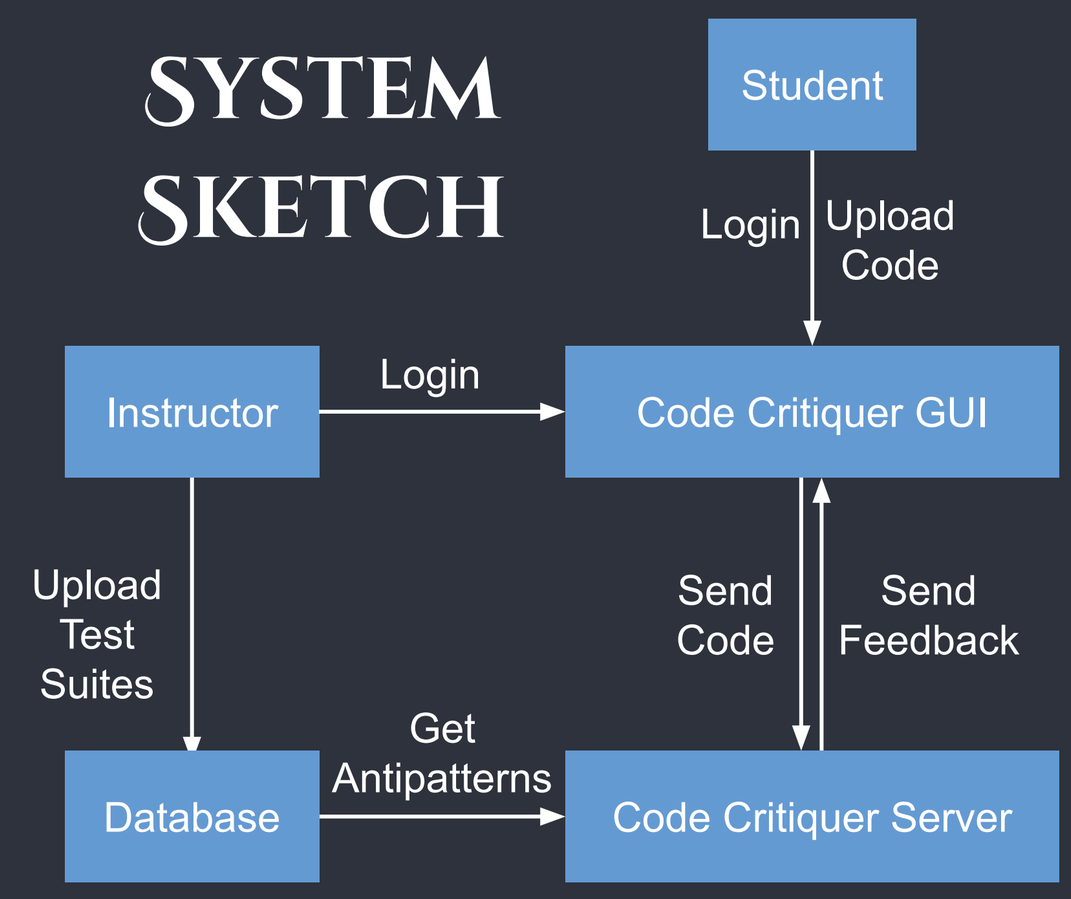
EXAMPLE ANTIPATTERN

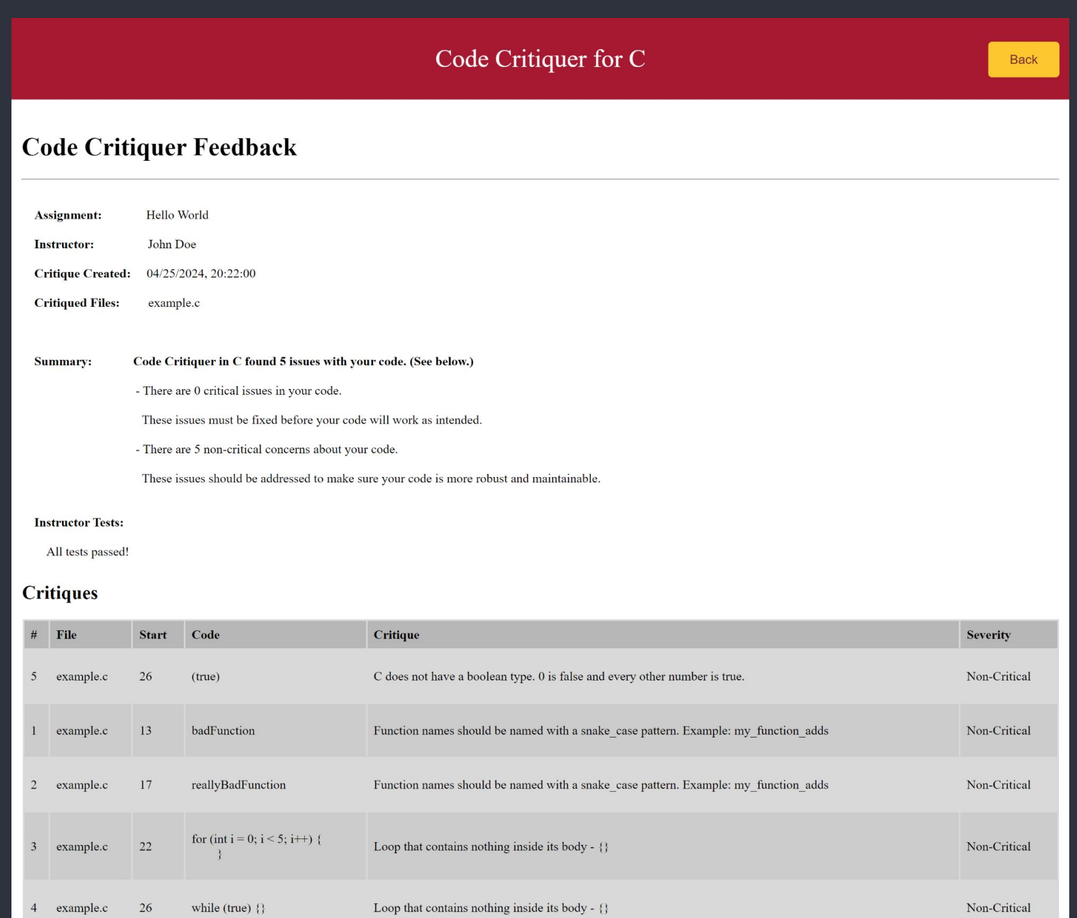
- Code:

```
while (i < 5) { }
```

- Message displayed to students:
  - "Loop that contains nothing inside its body - {}"
  - Also: file, line number, & code snippet
- Why does this help?
  - Something like the above code snippet, which would send the program into an infinite loop, isn't caught by the compiler and thus can be very hard for students, especially novice ones, to find.

SYSTEM SKETCH





Code Critiquer for C — Back

Code Critiquer Feedback

Assignment: Hello World
Instructor: John Doe
Critique Created: 04/25/2024, 20:22:00
Critiqued Files: example.c

Summary: **Code Critiquer in C found 5 issues with your code. (See below.)**
- There are 0 critical issues in your code.
  These issues must be fixed before your code will work as intended.
- There are 5 non-critical concerns about your code.
  These issues should be addressed to make sure your code is more robust and maintainable.

Instructor Tests:
All tests passed!

Critiques

#	File	Start	Code	Critique	Severity
5	example.c	26	(true)	C does not have a boolean type. 0 is false and every other number is true.	Non-Critical
1	example.c	13	badFunction	Function names should be named with a snake_case pattern. Example: my_function_adds	Non-Critical
2	example.c	17	reallyBadFunction	Function names should be named with a snake_case pattern. Example: my_function_adds	Non-Critical
3	example.c	22	for (int i = 0; i < 5; i++) { }	Loop that contains nothing inside its body - {}	Non-Critical
4	example.c	26	while (true) {}	Loop that contains nothing inside its body - {}	Non-Critical

Code Critiquer for C - Senior Design Project at Iowa State University
About

FUTURE PLANS

- Applying a C Compiler Preprocessor
  - Removes comments that currently can be identified as antipatterns
  - Applies macros that could fix antipatterns
  - Applies headers that allows for more antipattern identification
- Default Set of antipatterns
  - Provides better basic coverage for new instructors
  - Reduces demand for instructors to create their own
- Instructors share antipatterns
  - Instructors do not have to reconstruct antipatterns others have created