Detailed Design

Code Critiquer System for the C Language and Embedded C

sdmay25-23:
James Joseph
Samuel Lickteig
Alix Noble
Andrew Sand
Owen Sauser

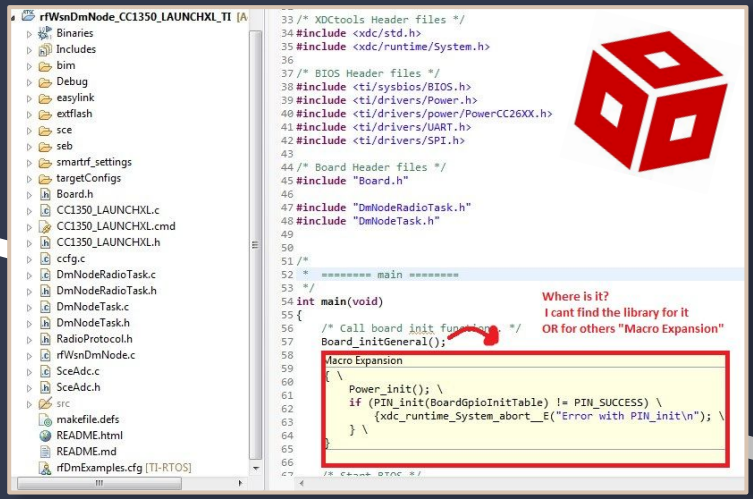
Code Critiquer System for the C Language and Embedded C

Project Overview

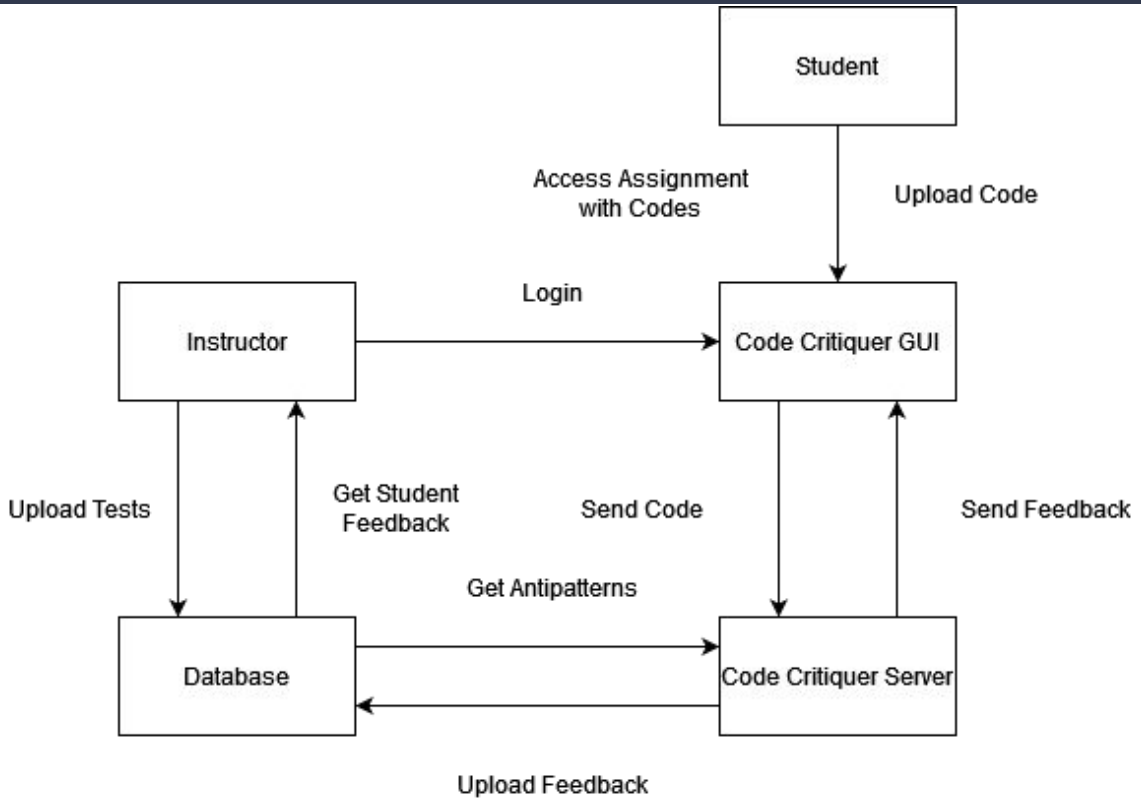
- Current state of project (continuation of sdmay24-34) is a web-based critiquer tool
  - Students upload C files to tool
  - Files are statically analyzed to search for antipatterns
  - Tool generates feedback
  - Students use feedback to improve skills
- Will modify current system and/or develop new prototypes
- Ideally tailored for CPR E 288
- Targeting a Spring Semester Prototype

Problem Statement

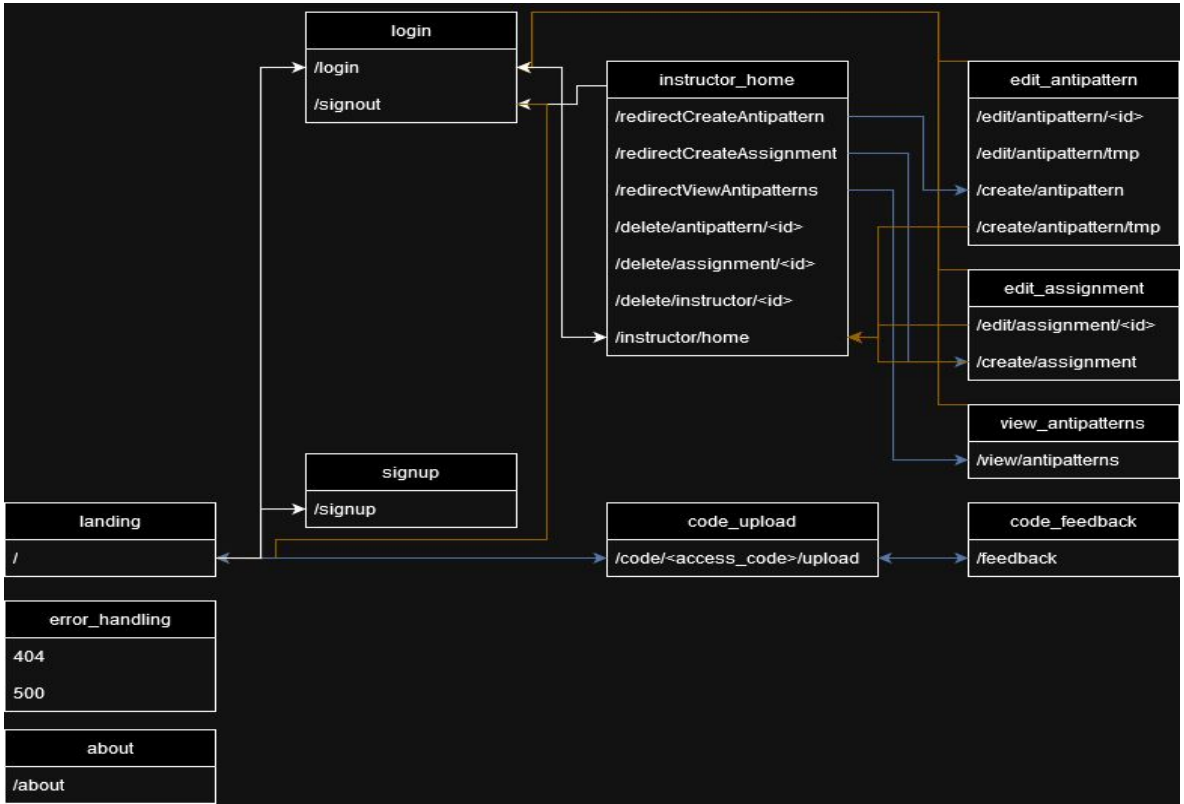
- Static Code Analysis is a challenging problem
- Many off-the-shelf solutions
  - Many leave a lot to be desired
  - They are not bespoke for CPR E 288 usage
  - No off-the-shelf “perfect combination” for what the project client needs
- Client needs a code critiquer that can...
  - Be accessed by students and instructors
  - Provide beginner-oriented feedback
  - Ability to give embedded and datasheet-focused feedback
  - Potentially integrate with Code Composer Studio



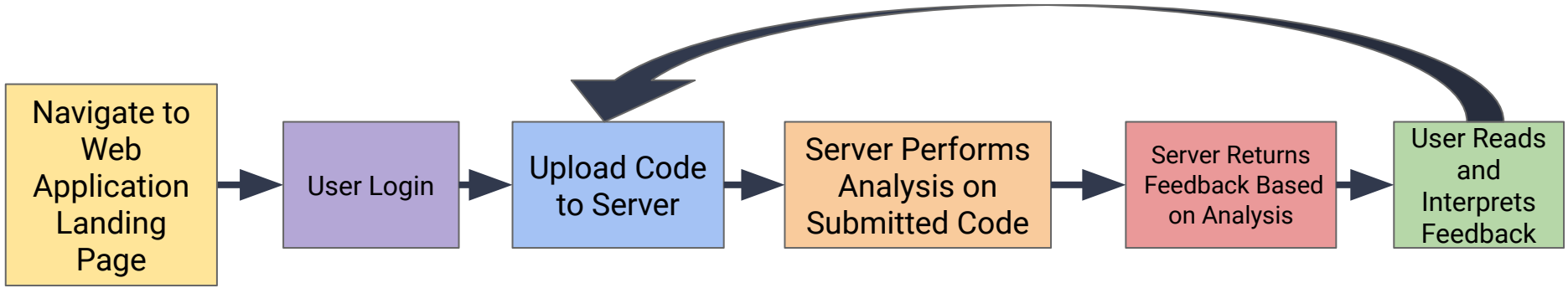
System Sketch



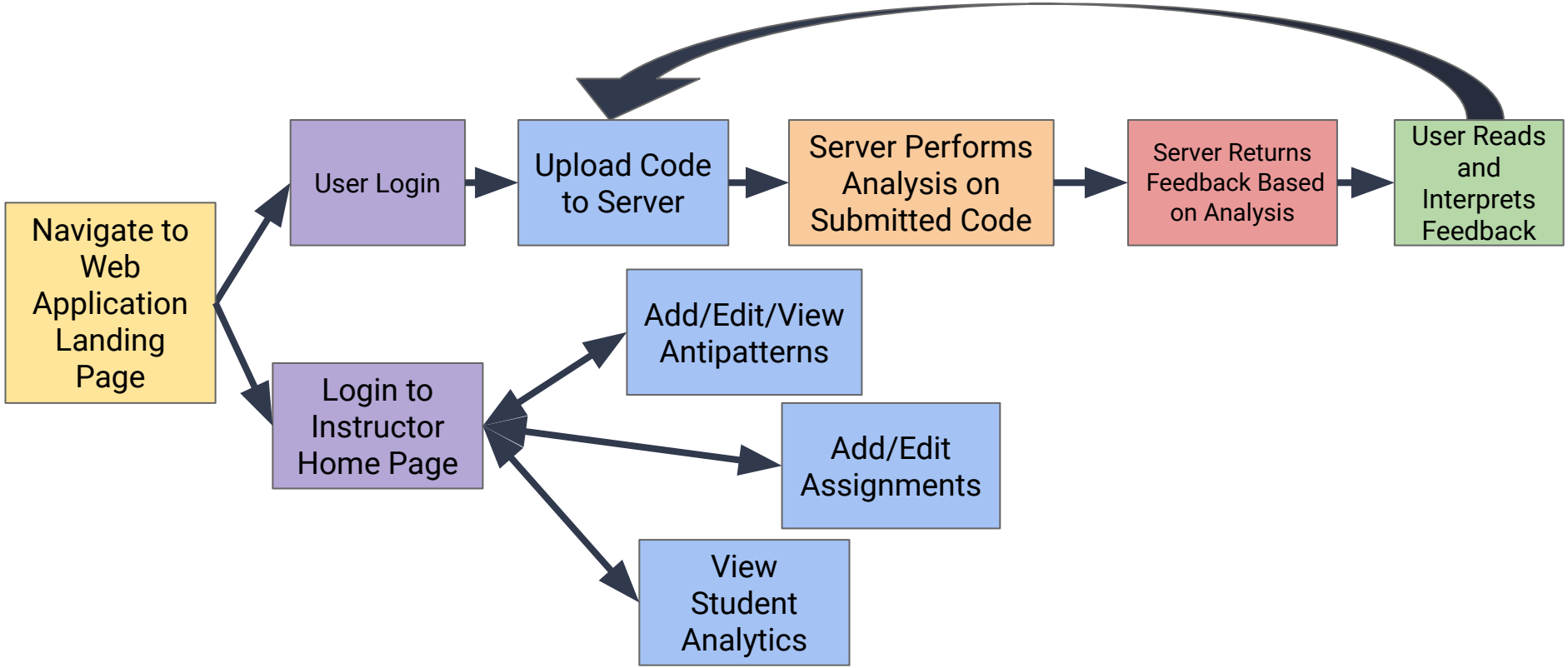
Page Flow Diagram



Functionality (Student)



Functionality (Instructor or Teaching Assistant)



Technology Considerations

- In practice this program needs to be put onto a server instead of hosted locally
- Whatever server it is put on must be able to handle potentially tens of CprE 2880 students using it and uploading code simultaneously
- Need to consider safe crash recovery options

Areas of Concern and Development

- Current design will fit user needs well
  - Provides students an efficient way of getting feedback on their code
  - Provides instructors and teaching assistants a way of tailoring feedback and analysis per CPR E 2880 lab
- Concerned about “tying it all together”
  - System has several complex parts that all need to work in unison to achieve desired results

Conclusions from Detailed Designs

- The project has a number of moving parts
- Critical to ensure that data flow is lossless between functions and pages of the system
- Users need to be able to navigate between each page without confusion
- Need to have a robust server for running the system

Any Questions, Suggestions, or Comments?