

Design Support Documents

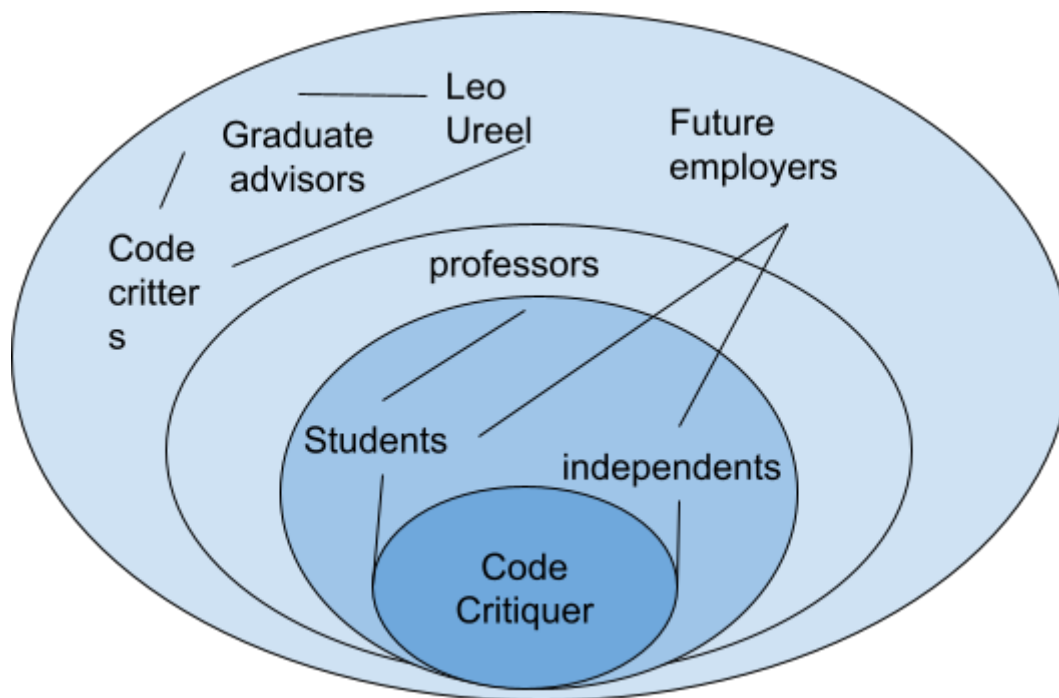
Andrea Lee
Team 6: Code Critters

System Description

The code critiquer is a web application which identifies errors and improvements within program files provided by users. This is accomplished through a database of common errors called “antipatterns”, which are compared to the provided code. The primary users for this application will be both students participating in university courses as well as independent users with no formal programming training.

Stakeholder Analysis

Onion Diagram



Stakeholders

Primary stakeholders

- Programming students – beginner programmers learning to code in a university
- Independent learners – users trying to teach themselves how to code

Secondary stakeholders:

- College professors – teaching programming classes, need code critiquer to assist with grading.

Tertiary stakeholders:

- Future employers – potential employers for primary stakeholders after graduation
- Dr. Leo Ureel – project lead for application project
- Code critters – design team for the project composed of undergraduate programmers
- Graduate advisors – usability testers for Code Critters made from human factors graduate students.

User Goal Table

Students	<ul style="list-style-type: none">● Feedback on code● Finding errors in code● Learning how to fix those errors● Avoiding future errors● Improved grades in courses
learners	<ul style="list-style-type: none">● Finding errors in code● Learning how to fix those errors● Avoiding future errors● Learning coding through correcting errors
professors	<ul style="list-style-type: none">● More efficient grading● Additional resources of students to improve their code● Assistance with finding and correcting errors they might otherwise miss
Future employers	<ul style="list-style-type: none">● Skilled employees● Increased productivity
Code Critters	<ul style="list-style-type: none">● Grade in class● Programming experience
Graduate Advisors	<ul style="list-style-type: none">● Grade in class● Usability testing experience
Leo Ureel	<ul style="list-style-type: none">● Finished application available for use● Help all other stakeholders achieve their goals

Programming students are using this application for feedback on their code before it is entered for a grade. They intend to find errors within their code, learn how to resolve these errors, and avoid them in the future. This process should also give them a better understanding of the code

Like the programming students, independent learners are trying to find, correct, and avoid errors in their code. However, without the assistance of the professor, these learners will try to learn programming through a process of trial and error facilitated by the application.

Unlike the primary users, professors will not be providing their own code to the application. Instead, they will either be providing the code of their students to assist with grading. They may also provide the application to their students to use as an additional resource to improve their code.

Future employers will likely never see the application itself. However, they may still see its benefits. The improvements in programming primary users will continue to be important even after their training is complete.

The Code Critters is the team of undergraduate students tasked with creating the application. Along with the graduate students working as usability advisors, are working on this project for a grade in their respective classes. While the undergraduates of the Code Critters are working to gain programming and application design experience, the graduate students are working for usability testing experience.

Leo Ureel is the lead scientist for the code critiquer project. As such, his goal is to have a functioning application capable of accomplishing the goals of all other stakeholders.

Personas

Persona 1: Jane Doe

- Age:18
- 1st year Computer Science Major
- Has not witten code before starting degree program
- Link to application was provided by professor
- Uses application frequently throughout the coding process

Persona 2: Rufus Xavier

- Age:25
- Electrician
- Learning programming independently using online resources
- Writes code specifically to learn programming, attempting to learn through experimentation

Persona 3: Dr. Steven Hawk

- Age: 66
- Jane Doe's intro to programming professor
- Provided link to application to all students
- Will not be using the application himself

Persona 4: Dr. Thomas Attkins

- Age: 45
- Data structures professor
- No programming students using the program
- Using the application as an automatic grader for assignments

Hierarchical Task Analysis

1. Find errors in code
 - a. Student writes program file
 - b. Student completes program file
 - c. File is used as input for application
 - i. Program is analyzed for errors
 - ii. Errors are identified
 - d. Errors are reported to user
 - e. User corrects errors in code

2. Find errors in code of others
 - a. Receive program file
 - b. File is used as input for application
 - i. Program is analyzed for errors
 - ii. Errors are identified
 - c. Errors are reported to user

Plan 1-1: The user's code contains no errors. The sequence only performed once

Plan 1-2: The user's code contains errors. Steps b-d are repeated until no errors are present.

Plan 2: Several different files are used in sequence. No files are used as input twice

This application will have two major use cases. For the first use case, the user will be using their own code as input for the application. Depending on whether any errors are detected the user will either only use the application once, or correct the detected errors and use the corrected code as their next input. This process is repeated until no errors are detected. This will be the most common use case for primary users. The detection and correction of errors will

The second use case involves the code of others being used as input. Unlike the previous use case, errors will not be corrected after detection. This usWhile the process may be repeated with other files, the same file will not be used as input after errors are detected.