

Design Support Documentation

CS5760

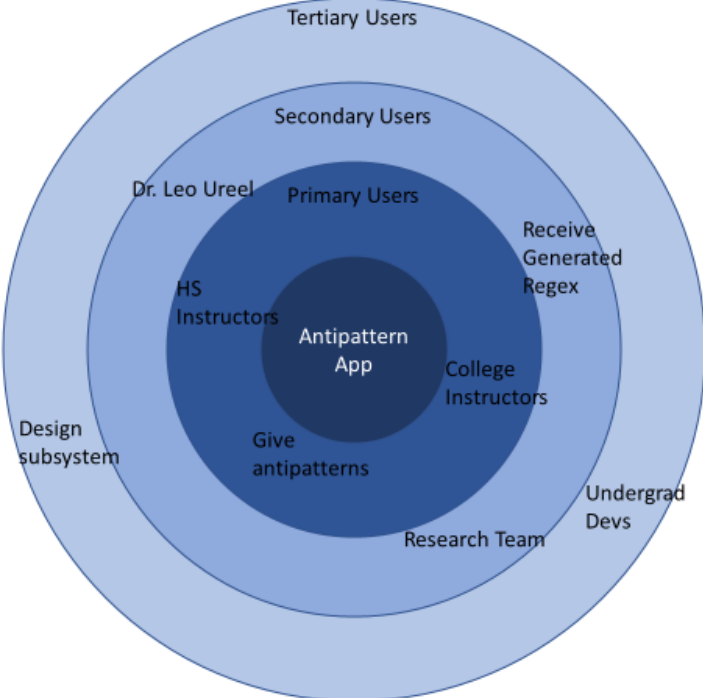
Laura Albrant

Application Summary

This is an application to replace the PatternDB application the client currently has with a much more user-friendly interface. In this application, the user provides incorrect code, highlights (in some form) the antipattern found within the code, and the software outputs a regular expression to go with that antipattern. Minimally, the app must be able to output the regex to a text file. However, if possible, it should insert the regex to the existing database. Lastly, an advanced mode to allow users with computer science, or regular expression, experience to directly input the regular expression for an antipattern that user found.

Stakeholder Analysis

Onion Model



Stakeholders

(Primary) Instructors

Instructors that one day will use this regex generator software in conjunction with Dr. Ureel's WebTA to provide their students an additional, automated feedback tool.

(Secondary) Dr. Ureel

A professor leading research that is in need of this regex generator software.

(Tertiary) Pattern Pandas

A group of undergraduate computer science students tasked with developing this regex generator software.

Stakeholder Goal-Influence Table

Stakeholder	Goal	Influence(s)
(Primary) Instructors Users	To give antipatterns to the overall, larger system.	Whether or not the instructor has a computer science background. What level of education the instructor teaches
(Secondary) Dr. Ureel His research team	To receive the generated regular expression for use in a separate part of the overall system.	The programming language the subsystem focuses on for input/regex
(Tertiary) Undergraduate developers	To design and implement this subsystem successfully.	Experience (or lack thereof) in various programming languages Knowledge of regular expressions Semester workload and general availability

Personas

Primary User Personas

Persona 1

Name: Dr. X

Age: 35

Residence: Michigan Technological University (Houghton, MI)

Job: Adjunct professor in the computer science department

Goal: Input antipattens into the system to build the database

Behavior: Eager to improve a database of regular expressions that will help students in the long run.

Relationship: Uses app to communicate with students and research team.

Persona 2

Name: Dr. Y

Age: 55

Residence: Case Western Reserve University (Cleveland, OH)

Job: Tenured professor in the computer science department

Goal: Input antipatterns into the system to build the database, use the database to teach students in their lab

Behavior: Interested in researching antipatterns and improving an overall shared application

Relationship: Uses app to communicate with students and lab.

Secondary User Personas

Persona 1

Name: John Doe (he/him)

Age: 19

Residence: Michigan Technological University (Houghton, MI)

Job: Undergraduate Research Assistant, College of Computing Student

John is a busy bee as both an undergraduate computer science student and a research assistant for Dr. Ureel. He is a part of the research team. His role is to program the MATLAB Code Critiquer/WebTA. John will connect the software that the Pattern Pandas team develops to the backend he is currently working on.

Persona 2

Name: Lauren Albrecht (she/her)

Age: 22

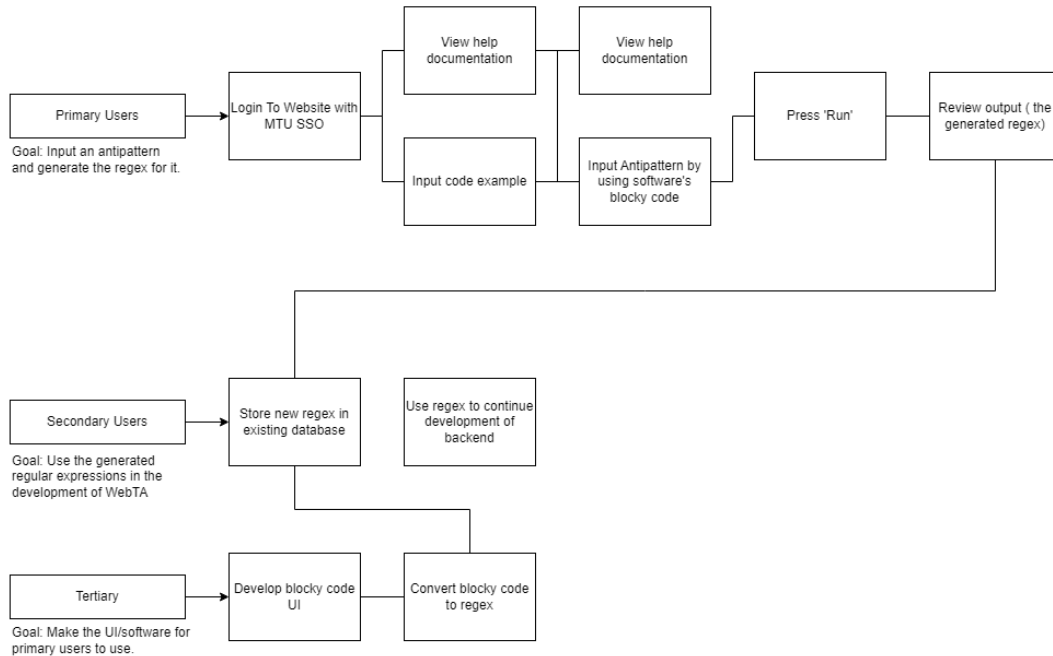
Residence: Michigan Technological University (Houghton, MI)

Job: Graduate Research Assistant

Lauren is a workaholic that leads the development team of Dr. Ureel's research and aids in development of the WebTA system. She will implement a subsystem related, but separate to software that the Pattern Pandas' are implementing.

Hierarchical Task Analysis

Diagram



Summary

Primary users have the goal of inputting an antipattern through the blocky code UI and reviewing the generated regular expression. If needed, they should be able to rely on help documentation to complete certain tasks.

Secondary users will have the goal of storing and using the generated regular expression(s) for their research and development of WebTA.

Tertiary users have the goal of developing this blocky code UI software.

Appendix: Meeting Notes

- Brainstormed to come up with the idea of having a Scratch-like, or blocky code, for the UI.
- The solution can be for only one language as long as the logic can be expanded for other languages (in the future by Dr. Ureel's research development team).
 - Knowing the grammar of a programming language should be very helpful in this blocky-code UI implementation.
 - Grammar changes between languages.
 - Scope for this project is tight, so as much as Dr. Ureel wants to explore all options, the undergraduate's need to rank and prioritize features.
- Features:
 - Needs to be behind an ISO login due to MTU policy.
 - Have an advanced mode (toggle on/off) to allow the user to tinker with the generated regex
 - Save the original generated regex so the user can revert their changes.
 - (Stretch goal) Show generated fail/pass test cases based on the regex
- Design:
 - Dr. Ureel mentioned Google's API Blockly. It seems very beneficial and time saving for them.
 - Focus on only keeping elements that pertain to the code example input, regex, and analyzing regex.
 - They need to ensure that they have helpful tips
 - Help documentation options:
 - Interactive (think of w3 schools)
 - Step-through walkthrough with screenshots