

Antipattern App Design

Supporting Documents

CS5760 – Human-Computer Interaction

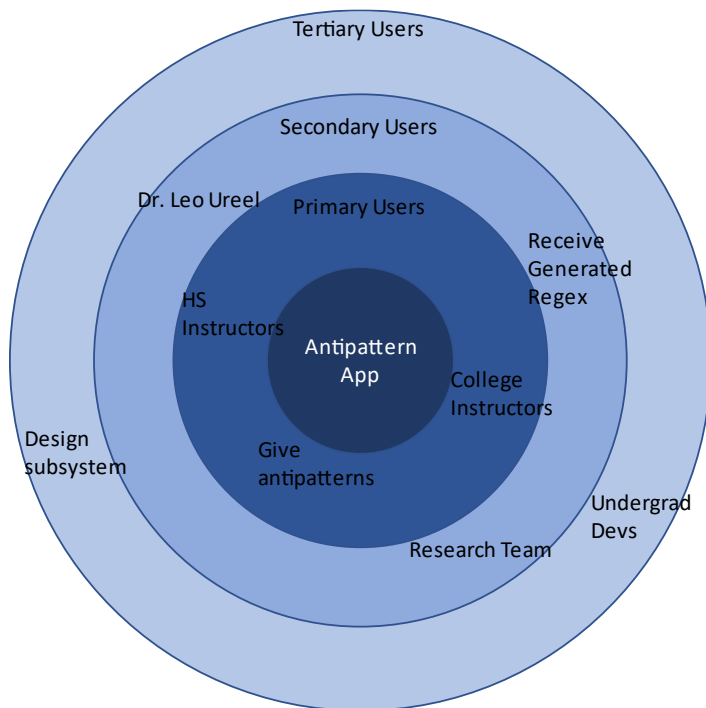
Brandon Woolman & Laura Albrant

App Description

The app must utilize regular expressions (code patterns) as the underlying tool. The purpose of this app is to make regular expressions easier to understand and to use. The app should have the ability to express or inform software of an antipattern. Can use a regex to spell-check code blocks (for the computer). In the simplest form: the user provides fallacious code (antipattern), the application makes a identifier (regular expression).

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

User 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 antipatterns 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

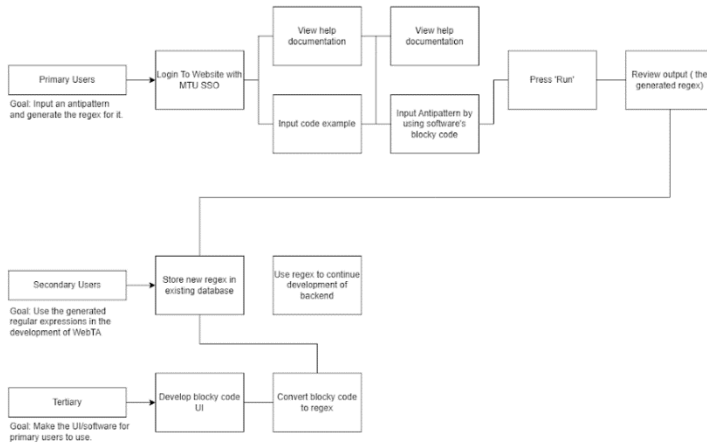
Residence: Michigan Technological University (Houghton, MI)

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 (Notes from 1/24/2023)

- Take some elements from Regex Generator
 - Assemble Regex blocks from an expression
- Tool tips – helps for first time (tutorial idea), which can be accessed from a tool bar later
 - Makes using the program a bit easier → Builds understanding
 - Leo likes *****
- Should language grammars be implemented
 - Leo: grammar should be fine if you're using one language
 - Must work for python or Java but easily convertible to other languages
 - Group wants it to be language agnostic
- Once Regex formula is generating. Can we make matching test that shows examples
 - HFE: That would be a stretch goal → Would be nice, must be understandable
- Should we have a login?
 - Leo: We should hide it behind a login to prevent hacking
- How do users submit their bad code?
 - Leo: Already being done. If you find a better way, feel free to implement
- What do we want the users to see? We know we want to see if test cases pass.
 - Leo: Since we don't really want them to tinker with it, hide it in the database. If there were high-level users maybe.
 - Hide the option to tinker, save the original
 - ^^ Prevents overload for basic users
- What are we wanting to store right away? Antipatterns and Regex? Language drop down too?

- Leo: Use the database to help generate some regexes. We don't need to know the languages, keywords or advice. Notes might be useful
 - Test cases are important
- Programmers could use Blockly