

App Summary 2

App Idea

The main purpose of the app is a Scratch-esque app with draggable blocks to help users with little to no knowledge of regular expressions to create regular expressions that identify Antipatterns (bad code formats) and upload them to a PatternDB. The app will also match against positive and negative test cases during construction. The app will mainly consist of a single, ideally non-scrolling, regular expression creation page.

The database is a secondary concern and should be largely copied from the existing collection or ignored and implemented after this semester.

Users

- Regex Authors
 - Teachers and professors, using the app to outline some bad programming behavior they have identified
 - May have little formal programming experience, app may be used by e.g. highschool math instructors using Matlab or even something like data entry
 - Likely have little to no experience with regular expressions

Workflow

- Login page: users will be greeted by a login page that uses MTU SSO login credentials, users will need to login prior to using the app
- Home page: users will enter their examples of good and bad code snippets and use blocks to construct regular expressions
 - Multiple steps, primary two are creating blocks and then combining them
 - Stretch/non-essential: Users familiar with regular expressions can access an “advanced” menu to start with an existing regular expression (or edit one created with the blocks) and blocks will be generated
- Stretch/non-essential: Regular expression showcase page: users will be able to view the detailed regular expressions broken down into blocks regular expressions
- Stretch/non-essential: Page will automatically generate a string that matches the regex as a sanity check that the regular expression is actually looking for the right thing.

Data

- Antipattern metadata - title, language
- Antipattern regular expression
- Antipattern matching and non-matching test cases

Anticipated Challenges

- Providing a comprehensive tutorial that does not overwhelm new users
- Balancing information presented to users and the complexity of regular expressions
- Learning and adapting Google’s Blockly framework to produce the regex blocks