

Cognitive Walkthrough Feedback

- If students or non-instructors are going to use the app, how will we have to account for that? Not all use cases will have test cases.
 - We also kind of forgot about where to put the “generated test cases.”
 - If users are trying to come up with a regular expression, they will have at least one test case.
- Consider some sort of warning about the block disconnection in addition to the red border on the canvas
 - Maybe tutorial can display what causes a red border
 - Perhaps an info icon or dismissable tooltip
 - Shows up the first time, lets you know the red border indicates that?
- Guide/tutorial for users who are lost while building the regex
- UX Questions:
 - Have we considered additional colors for colorblind users?
 - Red/green test cases is especially bad
 - Maybe we use contrasting colors in terms of saturation
 - Textured background - repeating slashes if failing?
 - Bolded if failing, lowered opacity if passing?
 - What are the tabs for on the prototype
 - Block library paging
 - What is the plan for the advanced mode?
 - Bring up a text box to edit the regex string
- Dr. Pastel's comment:
 - Library package for block like view?
 - Google block library blockly
- Leo Ureel's comment:
 - Team should translate 4 to 5 regular expressions to actual blockly. So that we can actually see what they look like and how we can implement it.
 - Possibly could do this alongside tutorial design
 - Possible stretch goal: Having an expert mode for more professional users or those who have used the application a lot: potentially removing labels that says function, repeat, text, etc. Allowing users to fit more blocks in one screen for longer regular expressions.
 - For longer regular expressions that won't fit in the existing screen: Figure out priority list to ensure that essentials and more innovative ideas are implemented
 - Don't integrate into patterDB so the team has more freedom with the database and not constrained by the existing website.
 - Leo like the expected and actual test case fields
- Feedback notes:
 - Potentially adding a landing home page for the purposes of the app.
 - Additional test case fields

- Methods of test case entering: plus button, editable field, add a new row as users enter a new test case
- Possibly tooltips for information button for more user help
- Export button functionality
 - Stretch goal: if team is able to get database and authentication working, we can have the regular expression uploaded to the database
 - If we don't have the time to implement a backend then the export will export text field in browser allowing user to copy and paste
- Error red outline alternative: display a red ⓘ mark that tells user what the error case is
 - Implementation must be colorblind friendly
- Dr. Wallace (Alloy student project) and Sarah Larkin have experience with Blockly
- Dr. Pastel's comments during Design Review meeting (2/16):
 - Create small groups for pair programming during implementation phase
 - Teams for blockly, css, test cases
 - Link cognitive walkthrough and feedback on home page
 - Pastel wants team to study blockly
 - Create a way to toggle and show regular expressions as the user edit the blocks
 - Change "Export" to something else that is more intuitive
 - "View RegEx"
 - Maybe blockly has a way to export the block structures itself for export
 - Document the language we are going to use
 - List constraints
 - Describe Purpose
 - Put this information in the GIT repository
 - Decision on no backend or database (no Grails)
 - Decision on whether or not to even bother with the login if we are not doing the database
 - Leo mentioned that PatternDB has the login page because there is a database behind PatternDB and he does not want anyone to mess with the MTU database. If we are not implementing database then we might not even need any login