05 Function Analyzer

Team Members

Ansgar Koopmann, Henri Truetsch

Basic Idea of the Component

- Show information about the funcation such as minima, maxima, intersection with y/x axis etc.

Categorization of the component

Categorize your component by highlighting (**bold**) your choice: Is it a

- **Local component** (e.g. a local password vault)
- Simulation of a cloud service (e.g. library service of your university library)
- Simulation of a team component, interacting with other teams (e.g. "Whatsapp")

Requirements

Must-Have

- Interpret user input as equation
- Calculate and display
  - minima
  - maxima
  - intersection with y/x axis
- Show y value of function for specific x value
- Find x value for specific y value

Nice-to-Have

- Calculate intersection of 2 fuctions
- Problem Generator
- Plot fuction inside a coordinate system

UI

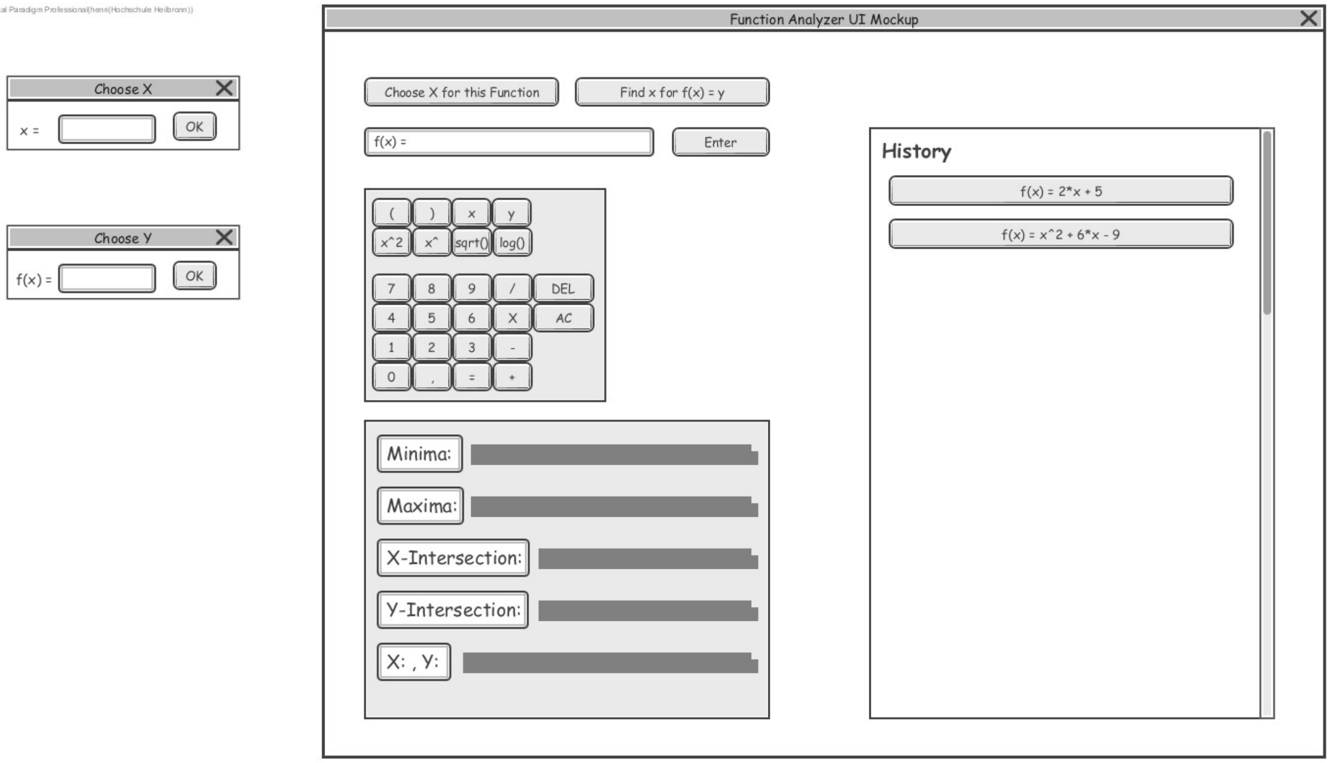
Categorize the UI by highlighting (**bold**) your choice: Is it a

- Demo-UI, to demonstrate aspects of the component (e.g. path finding demo)
- Test-UI to test aspects of the component (e.g. path finding with different weights)
- **Production-UI**, to use aspects of the component (e.g. card deck learning app)

---- End of initial description ---- Below here is for later use ----

---

Mockups



Function Analyzer UI Mockup:

User first enters a Function via the calculator buttons down below. Numbers and arithmetic operators(+, -, * , /) can also be entered using the keyboard.

The User confirms his input with the Enter button. Minima, Maxima, X- and Y-Intersection will be displayed automatically if they exist.

X, Y: will be display if the user enters his desired value via the button "Choose X for this Function" or "Find for f(x) = y". The buttons open the corresponding dialog as seen to the left.

As soon as a function is confirmed it will appear as a button in the History panel. If the user clicks on that button it will load into the "f(x) = " textfield.

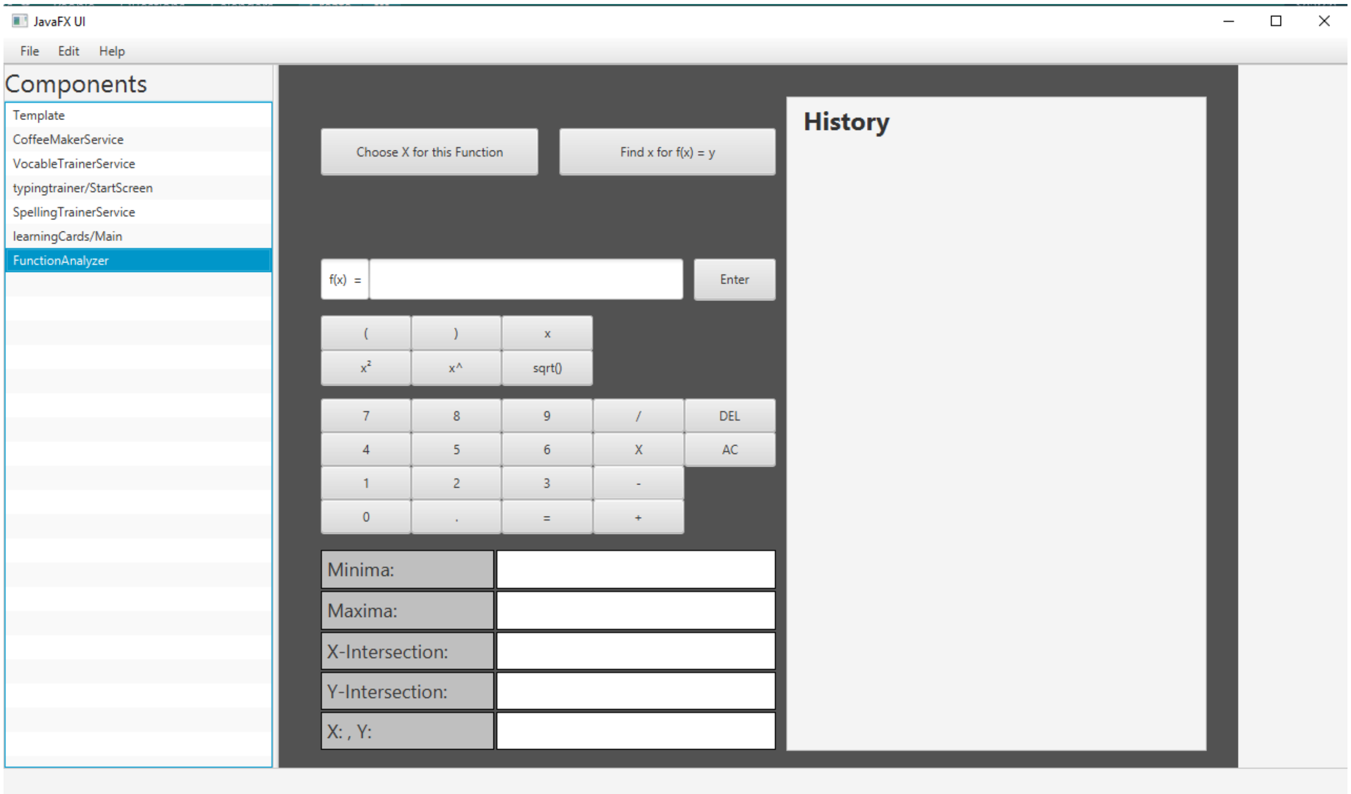
Identified Areas of UI Technology to Learn

On the basis of your mockups, derive which technology aspects you have to learn to realize your UI. E.g. if you have an item list in your mockup with the possibility to select multiple items (e.g. by checkbox), what do you have to investigate in the UI technology to implement that?

Based on the UI Mockup we have to learn how to:

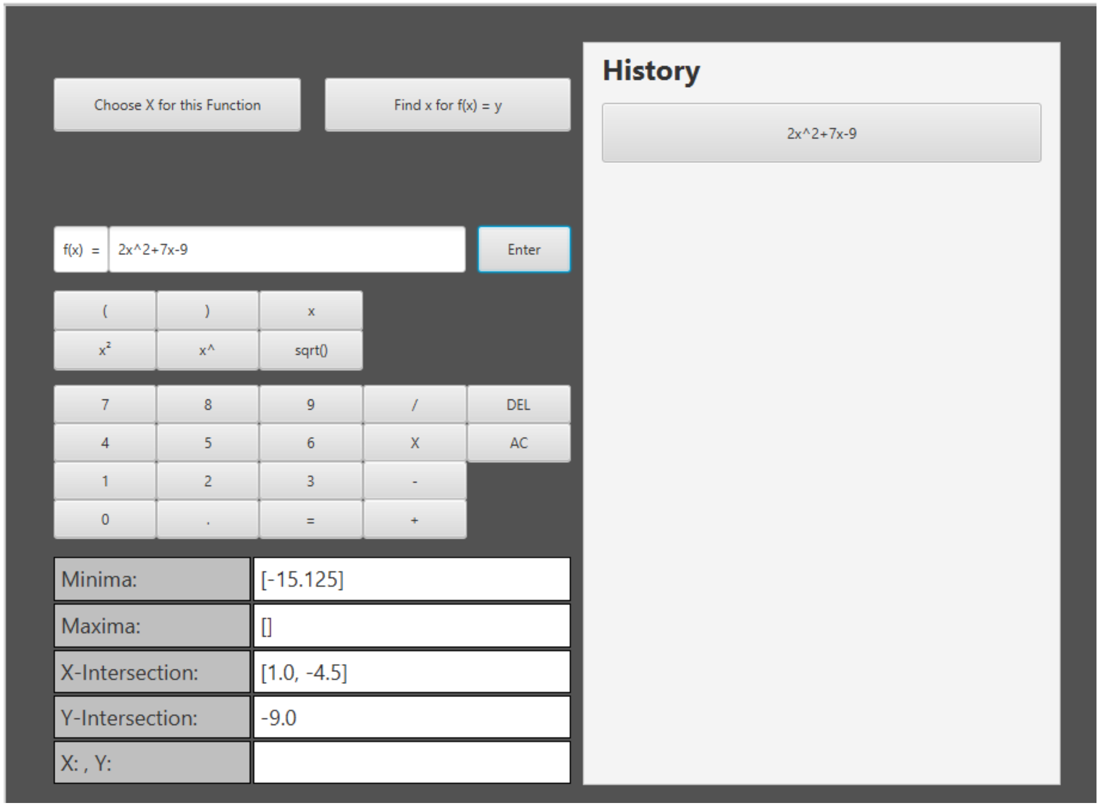
- Open new DialogWindows with press of a Button
- Take an input in form of a String and pass it on to the Implementation
- Parse a Button Value into a TextField
- Create a Button for entering a function into the TextField (History)

Screenshots



Here you can see the final GUI without any input. We had to take out some buttons for the calculator, like log() and y, because the functions would have become to difficult.

We also changed the background color from white to grey. The rest remains the same as in the UI Mockup.



f(x) = 2x^2+7x-9

Minima: [-15.125]

Maxima: []

X-Intersection: [1.0, -4.5]

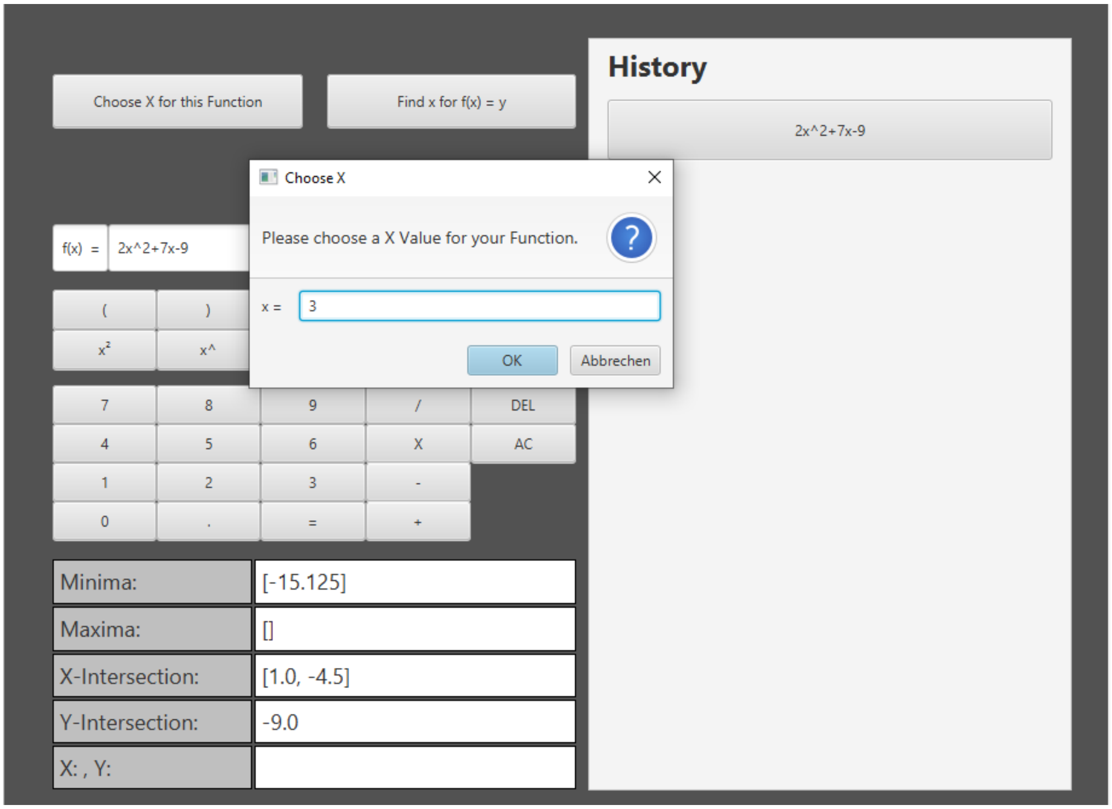
Y-Intersection: -9.0

X: , Y:

Here you can see the result for a function input. The result is displayed via the Labels down below. In the History on the right you can choose from different functions you already entered.

The maxima and minima can be calculated for third degree function, while X and Y Intersection can only be calculated for second degree functions.

As long as it meets the requirements of the calculator, the function input can have any complexity.



Choose X

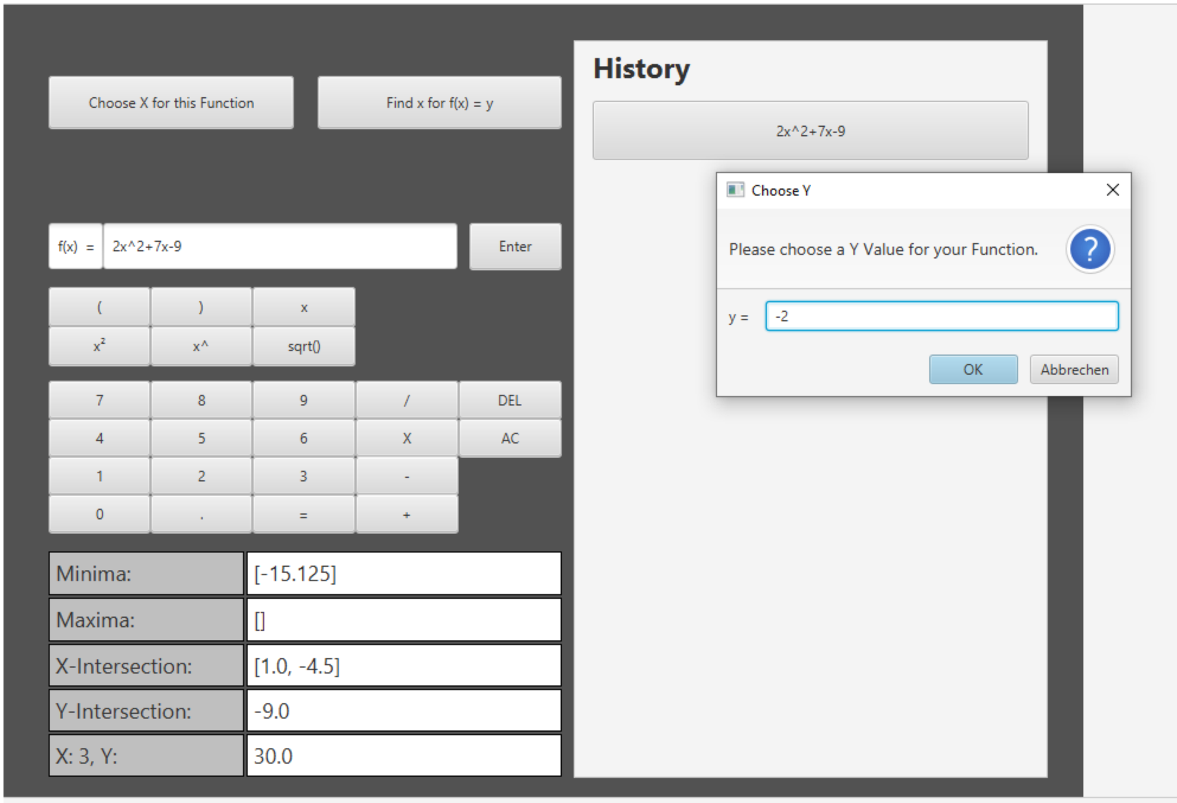
Please choose a X Value for your Function.

x = 3

f(x) = 2x^2+7x-9

Minima:	[-15.125]
Maxima:	[]
X-Intersection:	[1.0, -4.5]
Y-Intersection:	-9.0
X: , Y:	

Here you can see that the Choose X for this Function Button has been klicked on. A new DialogBox pops up to ask for a specific X value to be entered.



f(x) = 2x^2+7x-9

Choose Y

Please choose a Y Value for your Function.

y = -2

Minima:	[-15.125]
Maxima:	[]
X-Intersection:	[1.0, -4.5]
Y-Intersection:	-9.0
X: 3, Y:	30.0

The same logic applies to a specifc Y value. On the Button you can see the result value and initial X value input.

Lessons Learned

What was easy? What was time consuming? What was difficult to understand?

After getting over some initial difficulties with the screen builder, designing and implementing the UI was farily easy with the hardest part being to make sure that everthing behaves correctily when rezising the application.

The hardest part overall was by far the implementation. Mathimatical Fuctions can get very complex fast and have numerous special cases that result in different object states. Getting from the initial input of the function to a state that the function can be worked with proved to be the biggest challange.

Another time consuming part was writing the test to make sure the program would work with the aforementioned multitute of different cases, however UnitTest were also absolutely essential in making sure that changes to the program logic would not break other previously functional cases.