



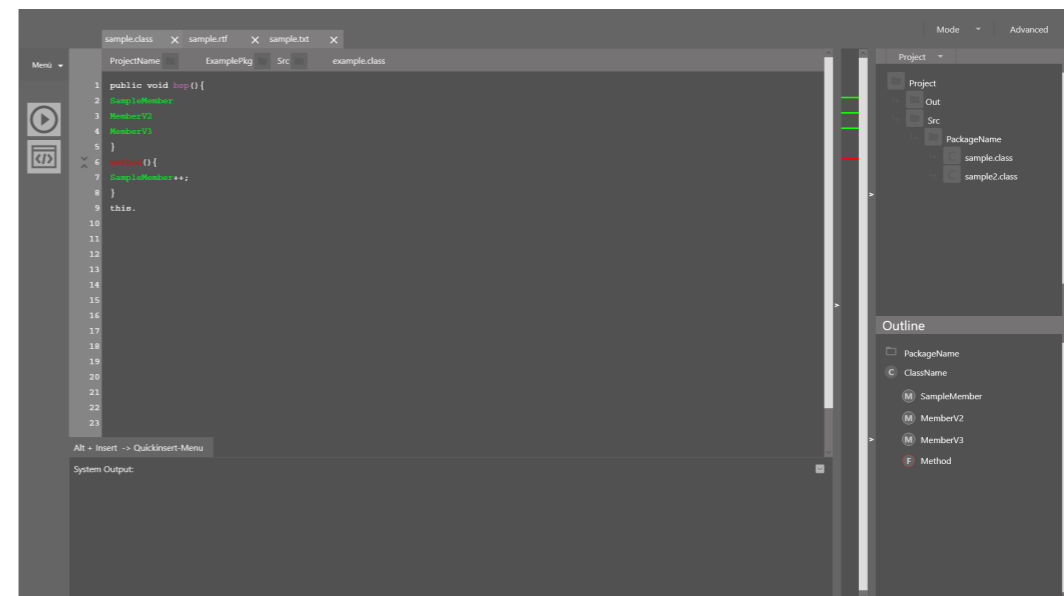
Notepad++ Redesign

In dieser Dokumentation möchten wir Ihnen den Prozess der Entwicklung des Notepad++ Redesigns aufzeigen, dabei auch die Bildung der Personas und desweiteren der Nutzergruppendefinition beleuchten.

Auf folgenden Seiten wird Ihnen nun dieser Prozess erläutert, über die einzelnen Personas, Konkurrenzanalysen, Features, die Ausarbeitung des Designs an Wideboard-Entwürfen, über den Clickdummy Prototypen, und Usability Tests bis hin zum finalen Design.

Das Notepad++ ReDesign ist die Verbindung zwischen professioneller IDE und simplen Text- sowie Source-Code-Editor mit einer benutzerfreundlichen, anregenden Programmierumgebung sowohl für Anfänger als auch Experten.

Vison Statement



Personas und wie wir sie eingesetzt haben:

Wir haben insgesamt drei Personas entworfen die jeweils komplett unterschiedliche Interessen und Fähigkeitsbereiche vertreten. Im Endeffekt konzentriert sich das ganze Projekt aber auf zwei der drei Personas, da einer der entstandenen Personas nicht mehr zu unserem Vision Statement passt.

Als primary Persona dient, Simone Meister eine Anfängerin in Sachen Programmieren und IDE's welche als angehende Informatikerin mit unterschiedlichen Programmiersprachen arbeiten muss und folglich auch unterschiedliche IDE's benutzt. Unser secondary Persona ist Victor Kowalczyk, ein professioneller Programmierer und Teamleiter in einer Firma die sich mit Webentwicklung beschäftigt. Sein Nutzen für Notepad++ könnte darin bestehen, dass er mit einem Programm alle Programmcodes seines Teams abrufen kann und so effizienter Probleme klären und Verbesserungen anwenden kann. Zuletzt haben wir noch eine dritte Persona mit Namen Leon Schreiner, ein Abiturient mit Schwerpunkt Informatik, welcher wenig bis gar kein Interesse am Programmieren hat, was er aber erst über die Schulzeit festgestellt hat.

Wir haben uns überlegt, ob auch so jemanden für das Redesign in Frage kommt, um ihm durch unser Redesign einen leichteren Einstieg

in die Programmierung geben zu können. Aber, wie weiter oben schon erwähnt, durch unser Vision Statement und den Werdegang unseres Projekts haben wir uns dafür entschieden, diese Persona komplett zu vernachlässigen und uns auf die anderen beiden zu konzentrieren.

Durch die Erstellung und die Arbeit auf Grundlagen der Personas konnten wir uns, nach anfänglichen Schwierigkeiten für einen Schwerpunkt festlegen. Bis dann aber ein endgültiger Use-Case entstanden ist, ist doch noch einmal eine gewisse Zeit vergangen. Im Endeffekt wird versucht, eine gewisse Balance aus beiden Personas, also Simone Meister und Viktor Kowalczyk, zu erreichen. Somit wird das Redesign schwerpunktmäßig auf Anfänger und zusätzlich auf den Stand von Profis gesetzt. Somit ist dann das System mit den Workspaces entstanden. Im Anfänger-Modus wird vor allem darauf geachtet, dass nur die wichtigsten Funktionen zum Einsatz kommen und Hilfestellungen wie Auto-Vervollständigung im Fokus sind. Dinge wie die Mini-Map oder ein Package-Browser werden dort also nicht eingeblendet. Es kann aber einfach per Mausklick auf den Workspace zum Profi Layout gewechselt werden um somit schnell auch auf die weiteren Funktionen Zugriff zu erhalten.

Personas



1

SIMONE MEISTER

24, Studentin Angewandte Informatik

“Gerade als Anfänger ist es gut, nicht gleich von Funktionen erschlagen zu werden, sondern Schritt für Schritt zum Ziel zu gelangen. Leichten Zugriff auf unterstützende Funktionen führen schneller zum Ziel”

PRIMARY PERSONA



2

LEON SCHREINER

16, Abiturient

“Programmieren klingt zwar Interessant, aber die Komplexität der Programme die zum Coden und Code-Verwalten zuständig sind, sind nicht sehr ansprechend”

OMITTED



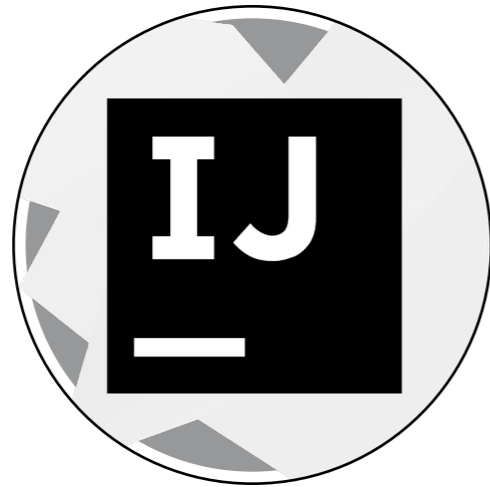
3

VICTOR KOWALCZYK

40, Teamleiter Webentwicklung

“Als Teamleiter einer nicht ganz unbekanntem Firma, die in der Webentwicklung arbeitet, besteht meine Aufgabe darin, Teams einzuteilen und zu leiten, damit diese schnell und effizient an Projekten arbeiten können.”

SECONDARY PERSONA



K O N K U R R E N Z

Auf den nächsten Seiten werden die Konkurrenzprodukte analysiert, auf ihre Features und Layouts, dabei kristallisiert sich bei allen ein ähnliches System heraus, welche Bestandteile benötigt werden:

Coding Area: hier schreibt der User seinen Code, es muss eine Zeilennummerierung für das wiederfinden einzelner Codesegmente geben und Code wird gehighlighted, somit farbliche Aufbereitung. Hier kann der User sich einzelne Segmente hiden, so dass der Code übersichtlicher dargestellt wird

Rechts zu sehen, das Layout von IntelliJ Idea

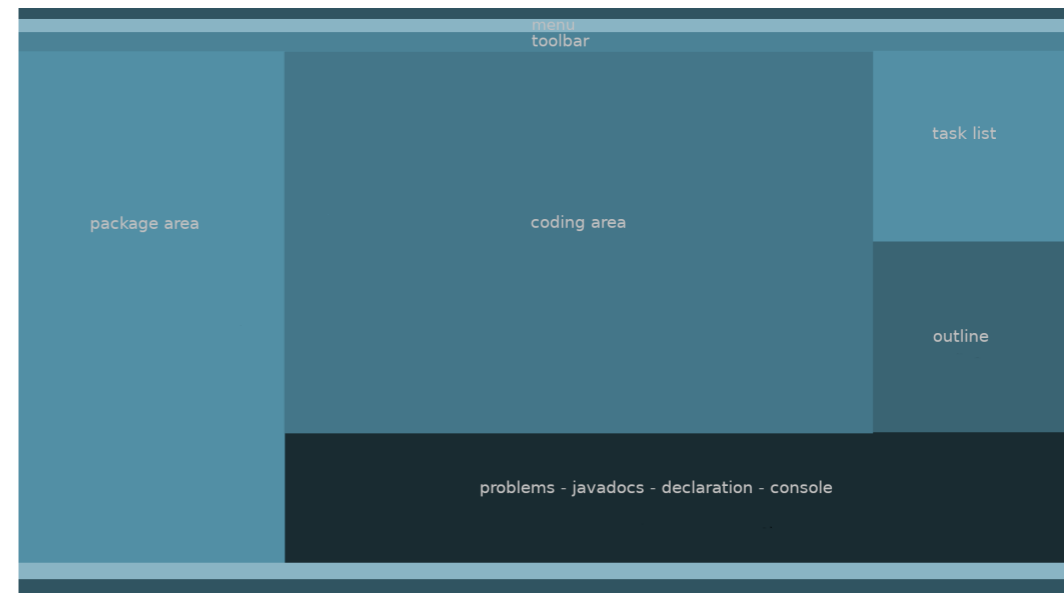
Console/Problems Multifunktionsbereich hier findet der User unter anderem die Textausgabe welche die Commandline Aus/Eingabe entgegen nimmt

Package Viewer: hierdurch kann der User in seinem Projekt und dem Filesystem navigieren, neue Projekte und Klassen anlegen

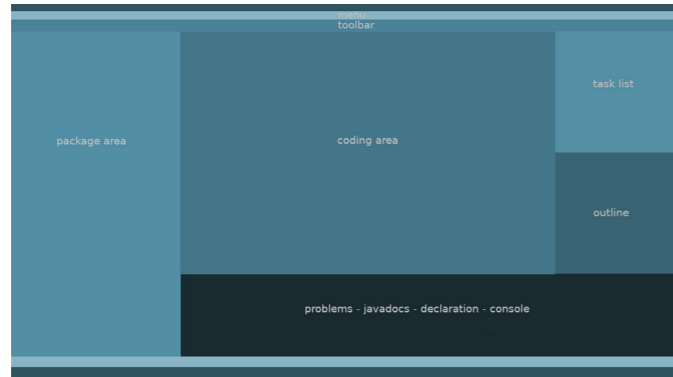
Outline: es wird dem User erleichtert Klassen und Variablen wiederzufinden

Codeminimap: Orientierung innerhalb der Dokumente

Taskleisten: für vielfältige Funktionsbuttons



Konkurrenz - Layout

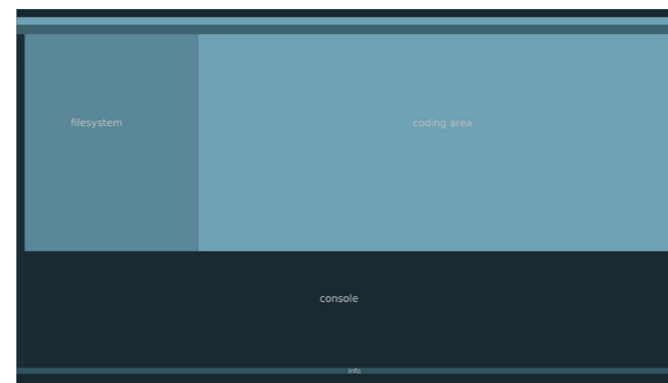


1

ECLIPSE

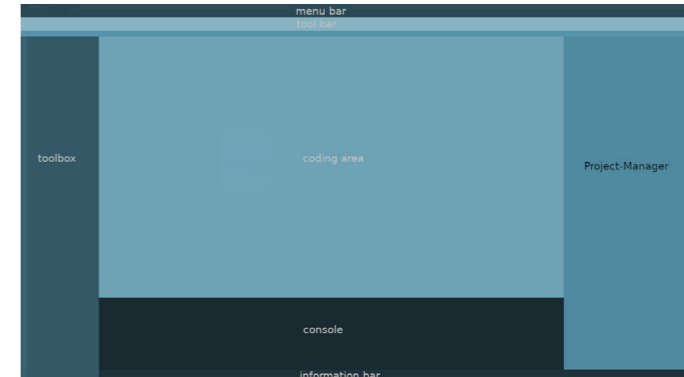
2

INTELLIJ

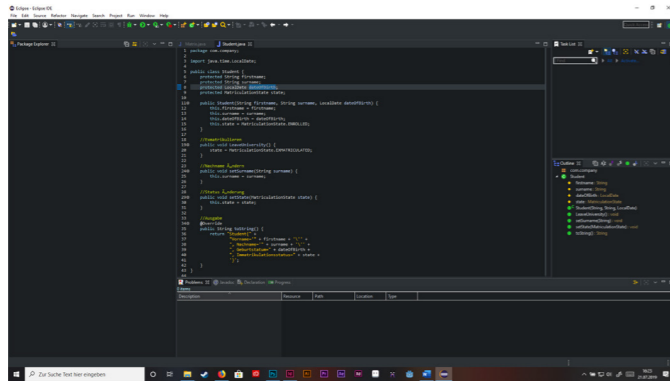


3

VISUAL STUDIO



Konkurrenz - Analyse



1

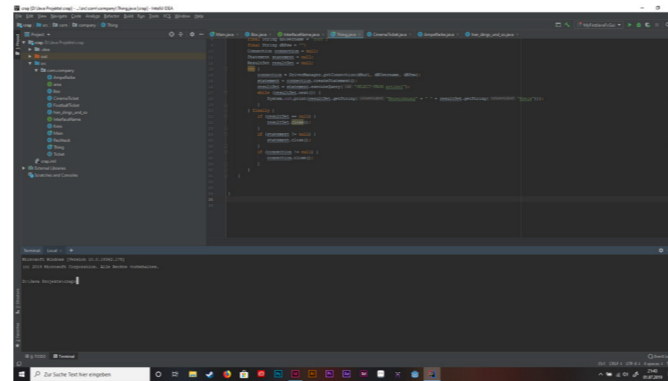
ECLIPSE

ECLIPSE FOUNDATION TION

Umfangreiches Feature-Set

Umfangreiche Features,
ausführlich dokumentiert,
Opensource

[HTTPS://WWW.ECLIPSE.ORG/](https://www.eclipse.org/)



2

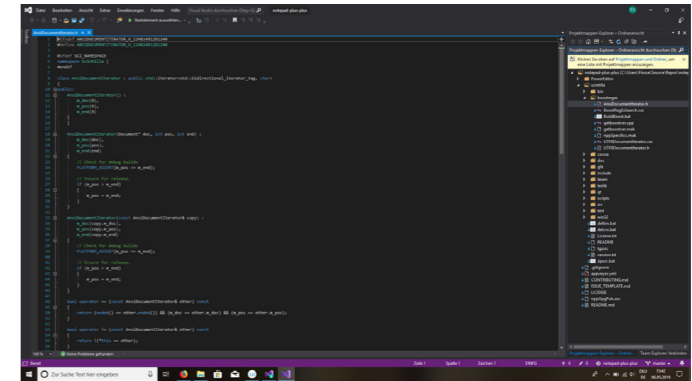
INTELLIJ IDEA

JETBRAINS

Ausführliche Hilfen/Tools

Innovative Hilfsfunktionen,
dem User wird hier viel Workflow-Optimierung geboten,
Komplex

[HTTPS://WWW.JETBRAINS.COM/IDEA/](https://www.jetbrains.com/idea/)



3

VISUAL STUDIO

MICROSOFT

Vorteilhafte UI

Übersichtliche UI,
sinnvolle Hilfsegmente,
Gut sortierte Funktionen

[HTTPS://VISUALSTUDIO.MICROSOFT.COM/DE/](https://visualstudio.microsoft.com/de/)

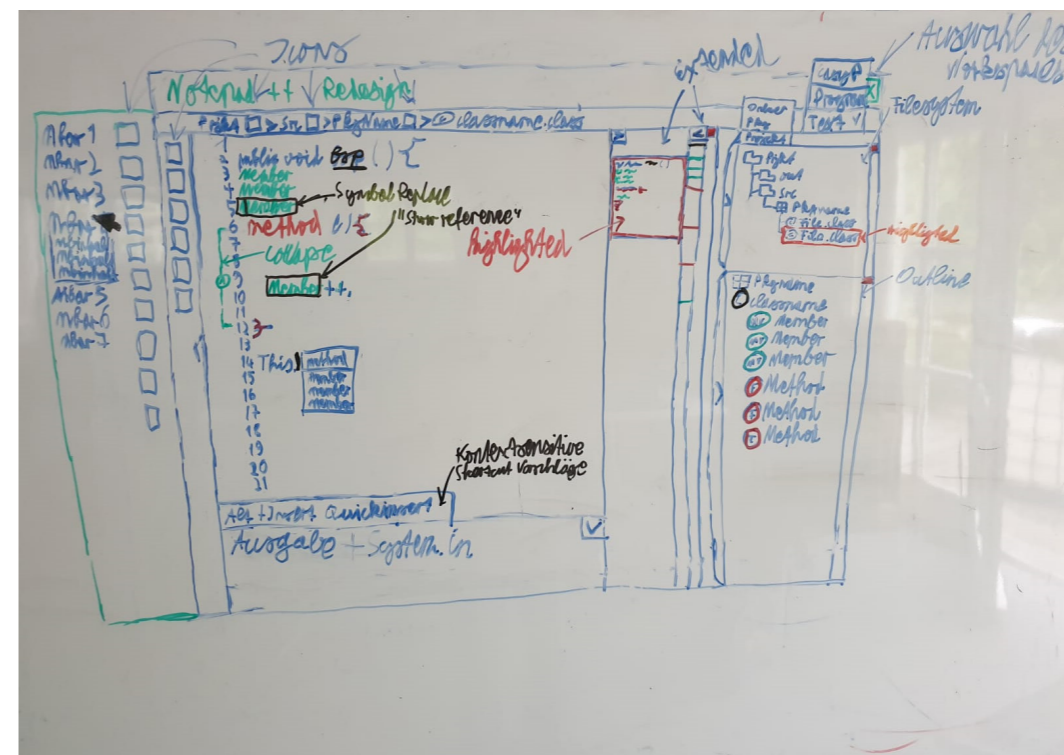
Wireframes

Die ersten Wireframes entstehen, noch auf dem Whiteboard, dabei werden Funktionsbereiche möglichst übersichtlich platziert, Tiefe geschaffen und Information hiding betrieben, Overview first, Zoom & Filter und Details on Demand nach Shneiderman beachtet.

Information Hiding wird durch dezente pfeilartige Symbole visualisiert.

Zoom und Filter erfolgt via Modi, der User kann sich die Informationsmenge und Sonderfunktionen anpassen, Stichwort Individualität. Es gibt die Modi Text, Easy und Advanced. Hier stellt der Textmodus eine Besonderheit dar, dieser reduziert die Informationsmenge auf die Coding Area.

Die Whiteboard Entwürfe wurden für den Advanced Modus entwickelt, somit sind alle Komponenten auf einmal zu sehen

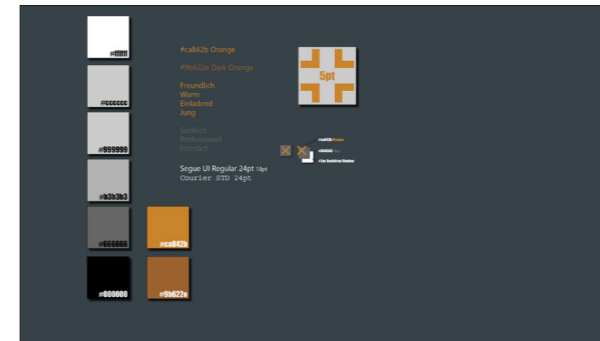


Whiteboard Entwürfe

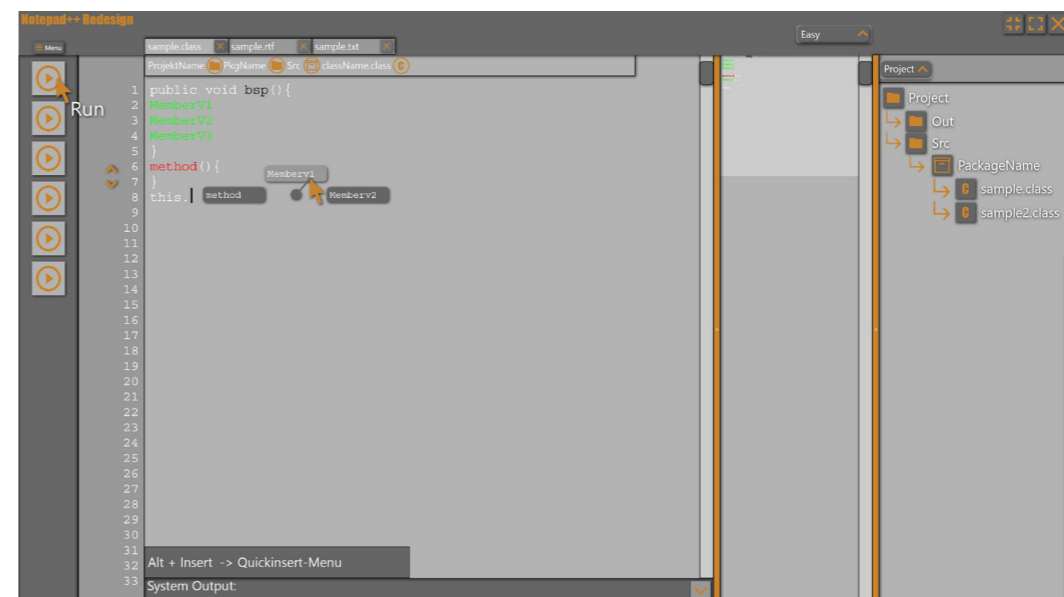
Wireframes

Es folgen digitale Wireframes, zu erst dem versatilen Rastersystem geschuldet in Indesign. Ein Style, der sich später durch User-Experience Tests als für längere Nutzung ungeeignet herausstellt wird entwickelt. Jener ist dadurch problematisch da ein zum Grau kontrastierendes Orange, in Verbindung mit konsequenten Schlagschatten für den User auf längerer Zeit unangenehm anzuschauen ist.

Digitale Hilfsfunktionen werden visualisiert und Buttons für den schon zuvor genannten Visual Information-Seeking Prozess eingebaut.



Prototyp des Stylguides



Indesign Entwurf

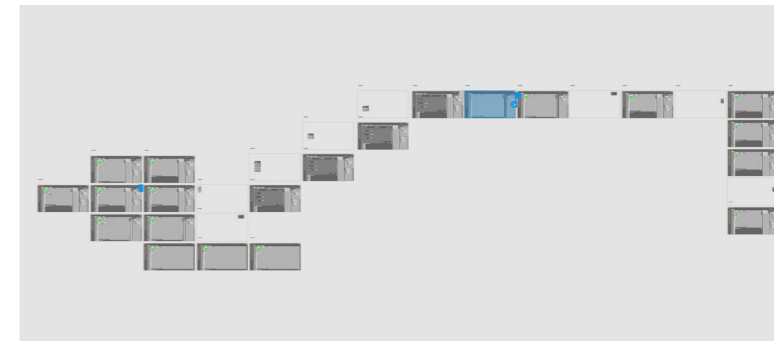
Clickdummy

Nun wird ein für User Experience Tests geeigneter Adobe XD Prototyp erstellt, dieser folgt dem finalen Styleguide, demnach kein akzentuierendes Orange, hier werden die Elemente des Indesign Prototypen auf ein flaches Design übertragen.

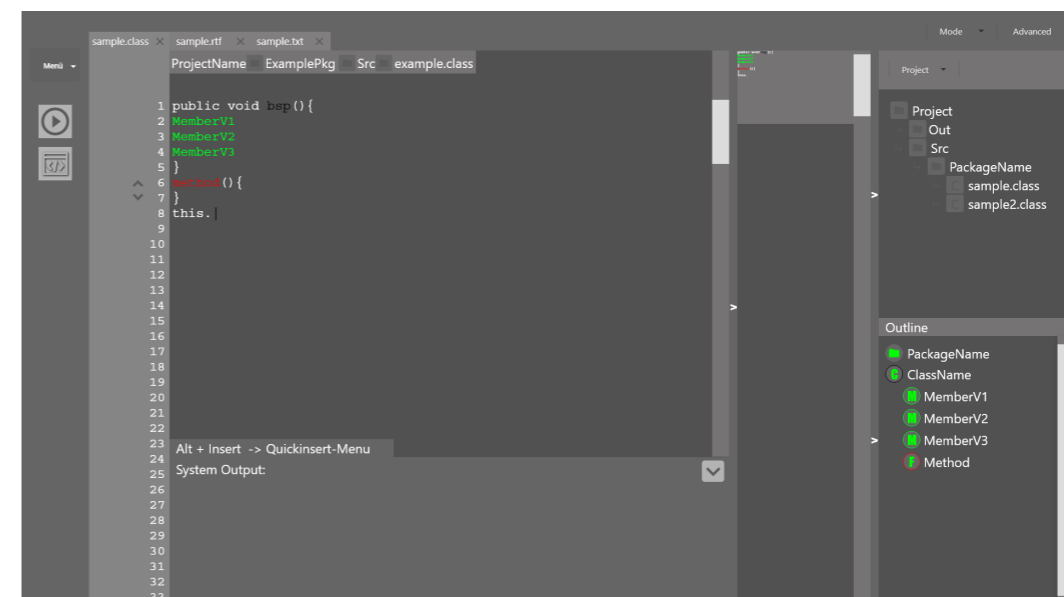
Sämtliche Buttons, Funktionen und Menüs werden auf das sleek Design angepasst, mehr Konsistenz wird geschaffen, Buttons werden nicht mehr durch übertriebene hervorhebung via Schlagschattten gehighlighted, sondern dank dem Gesetz der Erfahrung über bereits vertraute UI-Elemente, die an dem Google Material UI Kit orientiert sind.

Es gibt grundlegend zwei Methoden einen Clickdummy zu erstellen, die eine stellt ein lineares System dar, der User kann die einzelnen Screens nur auf einem Weg durchlaufen, dies ist für die Usability Test einerseits vorteilhaft, es kann gut nachverfolgt werden wo sich ein User befindet, jedoch kann dieser pro Screen nur einen Weg einschlagen somit ist die Problemquote schon im vorhinein verfälscht.

Ein geeigneteres System setzt auf einzelne lineare Abläufe die jedoch immer wieder auf einen zentralen Screen zurück kehren, somit kann auf diesem Screen jede Funktionskette einzeln angewählt werden, der User hat hier den vollen Funktionsumfang.

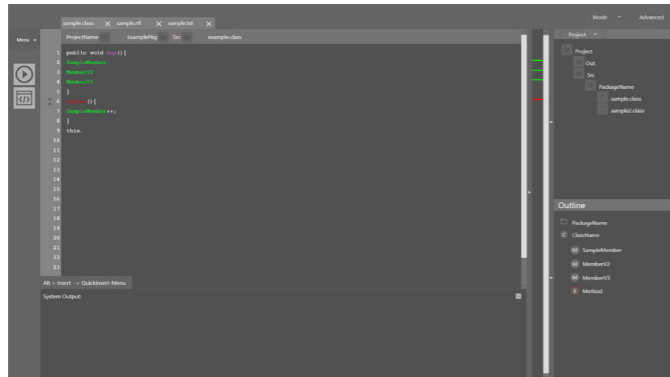


Aufbau des clickbaren Prototypen



Clickdummy

Finales Design

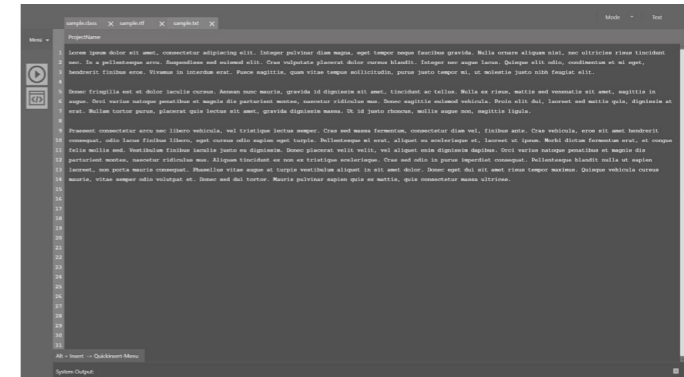


1

ADVANCED

2

EASY



3

TEXT

