



Przygotowywanie projektów graficznych

Zbiór porad i dobrych praktyk, które usprawniają tworzenie projektów graficznych.

Wstęp

Jako projektanci UI/UX wierzymy, że projekty które przygotowujemy są najlepszym rozwiązaniem problemów naszych klientów. Chcemy, aby efekty naszej pracy pokrywały się z wyglądem i działaniem finalnego projektu, a sam proces wdrożenia był maksymalnie przejrzysty i przyjemny dla wszystkich, którzy w nim uczestniczą.

Jak korzystać z tego dokumentu

Zdajemy sobie sprawę z tego, że tekstu jest sporo, a nasze projekty są tak różnorodne, że nie da się stworzyć dokumentu, który będzie dotyczył wszystkich idealnie.

Zacznij od przejrzania spisu treści, aby zlokalizować fragmenty które mogą dotyczyć Twojego projektu. Tam gdzie było to możliwe staraliśmy się podsumowywać paragrafy w pierwszym **wytłuszczonym** zdaniu, żeby ułatwić Ci przeglądanie i wyszukiwanie ważnych fragmentów.

Zgłaszanie błędów i sugerowanie pomysłów

Źródła tego dokumentu są dostępne w formacie **Markdown** pod adresem <https://gitlab.polcode.com/ui-ux/best-practices-frontend>. Zachęcamy do wysyłania PRóów z porawkami i pomysłami.

Przed przystąpieniem do projektowania

Spotkanie z ownerem

Mówiąc *owner* mamy na myśli dostępną nam osobę (klient, PM, inny członek zespołu), która posiada największy zasób wiedzy dotyczący tego projektu.

Naturalnie ilość informacji przekazywana podczas jednego spotkania będzie ogromna – nie wymagamy zrozumienia/zapamiętania wszystkiego. Omawianie projektu w ten sposób pozwala osobie przystępującej do projektowania zapoznać się z pełnym zakresem projektu i jego niuansami (np. cele biznesowe, logika biznesowa, ograniczenia, powody podjęcia pewnych decyzji, etc.).

Spotkanie z programistami

Przed przystąpieniem do projektowania spotkaj się z zespołem developerów, aby omówić podejście do projektu (np. jakiego typu urządzenia mają być wspierane, ilu oczekują responsywnych breakpointów, etc.), oraz z jakich technologii lub bibliotek planują skorzystać (np. Bootstrap, Material UI, D3, Chart.js, etc.). Pozwoli to zaoszczędzić czas i uniknąć rozczarowań podczas przekazywania ukończonych projektów graficznych.

Dobre praktyki projektowania

Format

Przekazując pliki lub dostęp do projektów w chmurze upewnijmy się, że developer (który w przyszłości ma na nich pracować) będzie w stanie je otworzyć.

Wykorzystujemy rozwiązania typu *cloud-based*, takie jak [Figma](#), [Adobe XD](#), [Sketch Cloud](#), czy [Zeplin](#) lub [Invision](#) (zamiast Photoshopa, którego uruchomienie np. na Ubuntu jest w najlepszym wypadku bardzo problematyczne).

Jeśli skorzystanie z w/w aplikacji nie jest możliwe, skontaktuj się z osobą, która przekonwertujemy projekty do formatu, który odbierający będzie mógł otworzyć na swoich urządzeniach.

Z Photoshopa korzystamy tylko i wyłącznie, kiedy klient tego wymaga, w innym wypadku zawsze wybieramy narzędzie, które jest stworzone z myślą o UI/UX.

Zanim odpalisz Figmę, Sketcha, czy Photoshopa

Skup się na celach biznesowych i rozwiązywaniu problemów. W początkowych etapach trwania projektu wyekstrahowanie informacji od klienta i przygotowanie architektury informacji są o wiele ważniejsze od docelowych kolorów i perfekcyjnie zaprojektowanych komponentów.

Zacznij od lo-fi. Projektowanie na papierze (np. Post-its), tablicy suchościeralnej, czy w narzędziach do budowy wireframe'ów pozwala na szybsze iteracje. Zmiany na tym etapie (w porównaniu z dalszymi, gdy większość kluczowych funkcjonalności będzie już zaprogramowana) są relatywnie tanie. Pamiętaj, aby wytłumaczyć klientowi, że to co widzi nie jest odzwierciedleniem ostatecznej jakości, a ma na celu oszczędzenie jego czasu i pieniędzy.

Organizacja projektu

Ekrany/artboardy powinny być uporządkowane. Decyzję o tym czy powinny być ułożone tematycznie (np. wszystkie ekrany związane z profilem użytkownika ustawione blisko siebie), czy logicznie (sekwencja ekranów odpowiadająca za flow płatności) czasami zależy od specyfiki projektu i pozostawiamy ją projektantowi.

Decyzję o tym czy ekrany responsywne (mobile, tablet, desktop) powinny być ułożone obok siebie, czy być pogrupowane rozmiarami na osobnych stronach pozostawiamy projektantowi i zespołowi – czasami ta sama osoba wdraża je wszystkie, a innym razem powstają odrębne aplikacje (np. web i React Native), które wdrażają różne osoby (i wygodniej jest im się obracać tylko w ramach swoich zadań). Najważniejsza jest tu wygoda zespołu i konsekwencja.

Dobrym zabiegiem jest dodanie dużych nagłówków (ok. 200pt) opisujących grupy ekranów – dzięki temu robiąc zoom out (aby zobaczyć wszystkie widoki) łatwiej odnajdujemy się w projekcie.

Pamiętajmy, aby ukończone ekrany dokładnie nazywać. Pomaga to w ich wyszukiwaniu i przeglądaniu. O ile w przypadku kilkunastu ekranów nawigowanie nawet po nieopisanych ekranach jest to łatwe, o tyle w przypadku projektów posiadających po kilkaset widoków konsekwencja w nazewnictwie zaprocentuje i ułatwi pracę wszystkim osobom zaangażowanym w proces tworzenia produktu.

Projektuj nowe widoki w miejscu z dala od już gotowych. W przypadku narzędzi pozwalających na kolaborację, staraj się wydzielić obszar zawierający już ukończone widoki (np. używając Pages w przypadku Figmy i Sketcha). Pozwoli to uniknąć nieporozumień które z ekranów są już gotowe, a nad którymi jeszcze trwają prace.

Uporządkuj skończone ekrany. Dobrą praktyką jest uporządkowanie warstw (nazwy opisujące ich zawartość, ułożenie ich w kolejności występowania na artboardzie) i grup (nazwy opisujące ich zawartość, usunięcie niepotrzebnych zagnieżdżeń) przed oddaniem ekranu do cięcia.

Pluginy ułatwiające ten proces:

- Figma: [Clean Document](#)
- Sketch: [Cleanup Useless Groups](#)

Interaktywne elementy interfejsu i przejścia pomiędzy ekranami. Jeśli program graficzny w którym przygotowujemy materiały nam na to pozwala (np. Figma, Sketch), warto poświęcić czas na zamarkowanie interakcji poszczególnych elementów interfejsu (np. kliknięcie w przycisk powoduje przejście do kolejnego ekranu lub pojawienie się pop up'u). Dzięki temu będziemy w stanie:

- Samodzielnie odkryć brakujące ścieżki użytkowników
- Dostarczyć klientowi *prototyp* aplikacji, co pozwoli mi na zweryfikowanie poprawności pierwotnych założeń i ewentualną szybką zmianę kursu zanim projekt trafi do wdrażania
- Dostarczyć programistom dodatkowy poziom informacji (w postaci interaktywnej dokumentacji), co zdecydowanie ułatwi im wybór.

Nie wszystkie interakcje można bez problemu wykonać i przedstawić w programie graficznym. Warto wtedy (w formie np. komentarza) podać link do interaktywnego rozwiązania, które jest online z przykładem, który w prosty sposób zobrazuje daną funkcjonalność.

Style, komponenty, grupy, frame'y i autolayout

Gdzie i kiedy warto je wdrożyć. Im bardziej Twój projekt się rozrasta, tym trudniej nad nim zapanować. W średnich i większych projektach kluczem do utrzymania porządku jest dyscyplina i korzystanie z dobrodziejstw nowoczesnych narzędzi (takich jak style, komponenty, grupy, warianty, frame'y i autolayout). W mniejszych (gdzie np. musisz dorobić tylko 2-3 ekrany do rozwijanej już aplikacji) nie warto tracić czasu na szczegółowe definiowanie komponentów – lepiej jest zdefiniować powtarzalne sekcje jako komponenty (np. header czy footer). W przypadku konieczności zmiany na tych elementach musimy to robić tylko w jednym miejscu (zazwyczaj pierwszy frame). Decyzję pozostawiamy Tobie.

Nie staraj się jednak wdrażać ich wszystkich na siłę już od pierwszych godzin projektowania. Takie podejście Cię spowolni i ograniczy możliwości iterowania. Zacznij porządkować (artboardy, grupy i warstwy) dopiero wtedy, gdy poczujesz, że projekt zmierza w dobrym kierunku, a z zaprojektowanych da się wyodrębnić małe powtarzalne elementy (np. przyciski, czy pola formularzy).

Inwestycja w style i komponenty oszczędzi Ci czas. O ile ich tworzenie bywa nudne i mozolne, to ostatecznie pozwoli Ci na utrzymanie spójności (te same elementy na przestrzeni całego projektu będą wyglądały tak samo), ułatwi wprowadzanie globalnych zmian (w przypadku gdy np. klient zażyczy sobie zmianę koloru wszystkich przycisków) i umożliwi szybkie tworzenie kolejnych ekranów (niczym budowanie z klocków LEGO).

Nie zostań niewolnikiem komponentów i wariantów. Fajnie jest mieć pojedynczy komponent z którego można zrobić input, pole tekstowe (z wieloma linijkami tekstu), czy dropdown, ale im więcej cech będziemy chcieli do niego dodać (etykiety, teksty pomocy, komunikaty błędu, stan hover, disabled, etc.), tym trudniej będzie nam je wszystkie pogodzić i uniknąć błędów podczas ich wykorzystywania. W tego typu przypadkach należy rozważyć wady i zalety różnych rozwiązań – czasami łatwiej, szybciej i wygodniej jest przygotować dwie osobne wersje komponentu zamiast walczyć z jego wariantami.

Nie ograniczaj się do tworzenia wyłącznie prostych komponentów. Z czasem zauważysz coraz więcej elementów układających się w większe, powtarzające się

moduły (np. grupy przycisków akcji pod formularzami, dropdowny, etc.), które będzie można wyekstrahować i okiełznać używając autolayoutu. Posiadanie dużych klocków (takich jak np. cały navbar, wraz z logo, linkami, avatarą i przyciskiem do wylogowania) pozwoli Ci nie tylko na szybsze tworzenie nowych ekranów, ale także na ich modyfikację z jednego miejsca (np. gdy klient zażyczy sobie nowego linku w menu) – dobrym punktem odniesienia jest metodologia Atomic Design. Tworząc większe komponenty (np. całe pole formularza, wraz z etykietą i dopiskiem z informacją pod polem) warto definiować je tak, aby składały się już z istniejących (mniejszych) komponentów, miały odpowiednio zdefiniowane odstępy i korzystały z auto layout (aby układ dopasowywał się automatycznie wraz ze zmianą któregośkolwiek z child-components).

Dodatkowe materiały:

- <https://www.figma.com/best-practices/groups-versus-frames/>

Grid

Mówiąc o gridzie mamy zwykle na myśli strukturę kilkunastu kolumn (a czasem też rzędów) o zdefiniowanych szerokościach i odstępach między nimi (w przeciwieństwie do ciągnącej się w nieskończoność siatki o jednolitych oczkach), w obszarze których będziemy starali się układać elementy interfejsu użytkownika.

Staraj się korzystać z wymiarów (liczba kolumn, szerokość kolumn, szerokość odstępów pomiędzy kolumnami) zdefiniowanych we frameworkach (np. Bootstrap, Material Design, czy Tailwind CSS), z których będą korzystali developerzy tego projektu – powinniście to ustalić na spotkaniu otwierającym projekt. Nawet jeśli developer nie zamierza korzystać z całego Bootstrapa, może on z łatwością wyciągnąć z niego sam grid i na nim oprzeć układ strony (pisząc resztę stylu samodzielnie).

Zaprojektuj swój grid. Jeśli powszechnie dostępne rozwiązania nie spełniają Twoich potrzeb/oczekiwań (np. gdy musisz umieścić na stronie banner o egzotycznych wymiarach, lub odtwarzacz wideo o konkretnym współczynniku proporcji), skonstruuj własny układ. Zalecamy oparcie go na układzie 12 kolumn (najpopularniejszy układ, ponieważ łatwo dzieli się na 1, 2, 3, 4, 6 i 12 równych kolumn), lub ewentualnie 16 kolumn (dzieli się na 1, 2, 4, 8, 16 równych kolumn). W sieci znajdziesz wiele tzw. Grid calculators, które w wizualny sposób ułatwią Ci znalezienie odpowiednich wymiarów kolumn i odstępów (tak, aby zmieścić się w najpopularniejszych szerokościach widoków responsywnych).

Trzymaj się gridu. Układanie elementów interfejsu wewnątrz kolumn ułatwi developerom pracę i pozwoli im na dokładniejsze oddanie Twojej wizji w docelowym produkcie.

Nie bój się sporadycznego łamania zasad. Ciągłe trzymanie się gridu może spowodować, że projekty będą “przewidywalne” i “nudne”. Aby tchnąć w nie więcej życia i dynamiki można czasami wyjść poza grid (np. grafiką, albo cytatem blokowym). Pamiętaj przy tym, aby tego typu zabiegi były przemyślane (by developer był w stanie je zreprodukować np. korzystając z pozycjonowania absolutnego i wartości procentowych) i robione ze smakiem.

Na stronie może być więcej niż jeden grid. Pamiętaj, że nie ogranicza Cię wyłącznie

główny grid. Każdy kontener (np. boczna kolumna, czy główna szpalta) może mieć swój wewnętrzny grid (np. jeśli główna kolumna składa się z 7/12 kolumn, nic nie stoi na przeszkodzie, aby te 7 kolumn podzielić znów na 12). Zwróć tylko uwagę, czy takie proporcje dobrze ze sobą współgrają wizualnie.

Grid horyzontalny. Poza zdefiniowaniem 12 kolumn (zwykle o stałej szerokości), warto jest dodatkowo włączyć powtarzające się w nieskończoność kolumny w trybie horyzontalnym (zalecamy paski po **8px** z odstępami **8px**, ewentualnie **4px/4px**, choć bardziej zaciemnia to obszar projektowania). Taki zabieg pozwala na łatwiejsze zapanowanie nad odstępami pomiędzy elementami i nadaje całości wertykalny rytm.

Po zdefiniowaniu bazowego rozmiaru fontu jako **16px (1em)**, developer może łatwo korzystać z okrągłych wartości relatywnych jednostek do definiowania odstępów (**8px=0,5em**, **32px=2em**, etc.).

Nowoczesne frameworki opierają swoje systemy odstępów właśnie na systemach **4px** (w Tailwind CSS **m-1** odpowiada margin: **4px**) lub **8px** (oczko siatki w Material Design), a korzystanie z ich wielokrotności ułatwi pracę developerom przy wdrażaniu tak przygotowanych projektów.

Dla ułatwienia sobie pracy, warto przestawić tryb dużego nudge (**SHIFT** + strzałki) z **10px** na **8px** – w Sketchu (Preferences → Canvas), a w Figmie (Preferences → Nudge amount).

Dodatkowe materiały:

- <https://bradfrost.github.io/this-is-responsive/patterns.html>
- <http://gridcalculator.dk/>
- <https://uxdesign.cc/responsive-grids-and-how-to-actually-use-them-970de4c16e01>
- <https://designsystemsrepo.com/design-systems/>

Typografia i teksty

Body text = 16px. Przy obecnych rozmiarach i rozdzielczościach ekranów przyjęło się aby ustawiać podstawowy rozmiar tekstu (tzw. body text, od rozmiaru fontu ustawionego dla elementu `<body>`) i nie schodzić poniżej tej wartości

Ustawienie mniejszego rozmiaru może spowodować problemy:

- niepotrzebny zoom-in na urządzeniach mobilnych (np. iPhone) po tapnięciu tekstu w takim rozmiarze,
- błędy w Google Search Console (mówiące o mało czytelnym tekście), co w dłuższej perspektywie może negatywnie wpłynąć na Page Rank strony.

Ogranicz liczbę fontów (i ich wag) na stronie. Korzystanie z wielu różnych krojów na jednej stronie obniża czytelność treści (wprowadzając chaos) i powoduje wydłużenie jej ładowania.

Unikaj skrajnych wariantów wagowych. Pomijając miejsca takie jak key visuelle, splash screeny, czy inne sporadyczne miejsca (w których chcesz osiągnąć zamierzony efekt wizualny oparty np. na wielkich literach o cienkich liniach), staraj się nie używać skrajnych wariantów wagowych fontów (takich jak thin, extrabold, czy ultra).

Zmieniając wagę fontu przeskakuj o 2 kroki. Aby uniknąć nieporozumień (czy to normal/medium, lub semibold/bold) pomijaj wagi i używaj par, które nie są sąsiadami, np. Extralight(200)/Normal(400), Medium(500)/Bold(700), lub Normal(400)/Bold(700).

Nie zmieniaj wagi dla stanu hover. W przypadku gdy stan normalny i hover będą miały różne wagi, wersja podświetlona kursorem stanie się dłuższa (efekt nasila się wraz z długością linku), co może spowodować “pływanie” sąsiednich elementów a nawet rozbicie układu strony.

Szpalta tekstu powinna mieć pomiędzy 60 a 80 znaków. Dłuższe linijki powodują trudności w śledzeniu tekstu wzrokiem (odnajdywanie początku kolejnej linii tekstu), natomiast krótsze zaburzają naturalną płynność rytmu czytania. Łatwym sposobem określenia optymalnej szerokości kolumny lanego tekstu jest trzykrotne wypisanie wszystkich małych liter angielskiego alfabetu (78 znaków).

Wyznaczanie górnej i dolnej krawędzi linii tekstu. Najwyższą z powszechnie stosowanych liter jest Å (U+00C5), natomiast najniższą jest y (pomijając znaki specjalne składane przy pomocy ligatur systemu Unicode).

Rytm wertykalny. W celu poprawy czytelności tekstu i uczynienia jego składu bardziej estetycznym warto zadbać o proporcjonalny rytm wertykalny. Pomaga przy tym włączenie poziomych kolumn (opisanych w sekcji Grid horyzontalny) i ustalenie wysokości linii wszystkich tekstów jako wielokrotności tego gridu. Teoria i przykłady:

- <https://zellwk.com/blog/why-vertical-rhythms/>
- <https://betterwebtype.com/articles/2018/10/15/rhythm-in-web-typography/>

Hierarchia nagłówków. Podobnie jak w przypadku doboru par wagowych, rozmiary poszczególnych poziomów pełnią bardzo ważną rolę w wizualnym uporządkowaniu informacji. Dobrze dobrane rozmiary i wagi poszczególnych poziomów spowodują, że użytkownik nie będzie miał wątpliwości na jakim poziomie treści się znajduje.

Wiele frameworków (np. Bootstrap, czy Tailwind CSS) posiada przemyślane i sprawdzone w bojach hierarchie nagłówków (i innych elementów typograficznych), które warto wykorzystać (choćby jako punkt wyjścia w przypadku zmiany fontu na niestandardowy).

Jeśli jest to uzasadnione, nie bój się modyfikować istniejącej hierarchii lub tworzyć własnej. Przydatnymi narzędziami będą następujące kalkulatory:

- <https://www.gridlover.net/try>
- <https://type-scale.com/>

Unikaj Lorem ipsum. Staraj się pracować na docelowych tekstach (a tam gdzie nie jest to możliwe, używaj zamienników – przykładem może być np. artykuł skopiowany z innej strony). Dzięki temu szybciej wyłapiesz ewentualne problemy i brakujące elementy (np. listy wypunktowane, czy cytaty blokowe).

Tłumaczenie tłumaczeniu nie równe. Projektując stronę, która ma pozwalać na przełączanie się pomiędzy różnymi wersjami językowymi, pamiętaj, że te same słowa w różnych językach mogą mieć różną długość (np. “FAQ” po francusku to “Questions Fréquentement Posées”, a “Cart” po niemiecku to “Einkaufswagen”, co może skutkować “rozjechaniem się” layoutu). Artykuł

<https://www.w3.org/International/articles/article-text-size.en> zawiera przykładowe “mnożniki” długości słów różnych języków.

Różne wersje językowe to nie tylko tłumaczenia. Wraz z obsługą tłumaczeń warto zadbać również o odpowiednie formaty zapisu walut, liczb, dat, adresów i pól formularzy (np. w przypadku płatności).

RTL to nie tylko odwrócenie kierunku tekstu. Osoby korzystające z języków takich jak Arabski, czy Hebrajski oczekują, że cała strona będzie “odbita lustrzanie”.

Dodatkowe materiały:

- <https://charactercounttool.com/>
- <https://www.fontpair.co/>

Dostępność

Wyróżnij interaktywne elementy. Wszystkie interaktywne elementy powinny być przejrzysto oznaczone (np. poprzez podkreślenie tekstu, dodanie cienia, użycie koloru akcentującego)

Zaprojektuj stan hover. Wszystkie interaktywne elementy powinny być dawać feedback po najechaniu na nie kursorem (np. poprzez wyświetlenie/ukrycie podkreślenia, zmianę koloru lub intensywności cienia).

Odpowiedni poziom kontrastu. Zgodnie z wytycznymi standardu **WCAG 2.1**, kontrast pomiędzy body text a tłem powinien wynosić conajmniej **4,5:1**, a dla nagłówków/większych tekstów **3:1**.

Etykiety pól formularzy powinny być cały czas widoczne. Nie ma przeciwwskazań przeciwko ich rozsądnemu animowaniu (jak ma to miejsce w przypadku Material Design). Nie powinno się ich natomiast usuwać/ukrywać, a w ich miejsce używać tekstu w placeholderach (ponieważ znika gdy tylko zaczniemy cokolwiek wpisywać w danym polu). Zadaniem placeholderów jest jedynie podpowiadanie użytkownikowi jakiego typu danych (i w jakim formacie) od niego oczekujemy.

Dodatkowe materiały:

- <https://webaim.org/resources/contrastchecker/>

Style guide

Omawiając to zagadnienie mamy na myśli dedykowany page lub artboard, gdzie znajdują się podobne do siebie komponenty (np. zestaw kolorów, hierarchia nagłówków, kontrolki formularzy, etc.). Pozwala to na łatwe wychwycenie ich wspólnych właściwości (np. podobieństwo fontów, obramowań i paddingów wewnątrz pól tekstowych, list) i utrzymanie spójności w projekcie. Osoba wdrażająca nasze projekty będzie wiedziała jak ma wyglądać zwykły input w różnych stanach (pusty, z treścią, z placeholderem, podświetlony kursorem, aktywny, zablokowany) i w różnych wariantach (zwykły input, input do wprowadzania kwoty w danej walucie, dropdown, etc.)

Dodatkowym atutem styleguide'u jest możliwość korzystania z niego jak z pudełka z klockami Lego – nowe ekrany tworzy się szybciej i pewniej (mając świadomość zachowania spójności z resztą projektu). W ekstremalnych przypadkach (np. nasza nieobecność) programista zaopatrzony w dobrze przygotowany style guide będzie sam w stanie zbudować brakujący moduł lub widok z posiadanych komponentów.

Popularne elementy

Nie jesteśmy w stanie przewidzieć i opisać tutaj wszystkich wariantów elementów jakich będziemy potrzebować (przygotowujemy projekty dla różnorodnych rynków i odbiorców), ale z doświadczenia wiemy, że jest kilkanaście typów, które powtarzają się na przestrzeni różnych projektów. Potraktuj tę listę jako inspirację w poszukiwaniu elementów, które mogły Ci umknąć podczas projektowania.

Must have – Jeśli opisywany element da się zastosować w przygotowywanym projekcie, jego przygotowanie jest wymagane.

Nice to have – Jeśli opisywany element da się zastosować w przygotowywanym projekcie, jego przygotowanie jest mile widziane.

KOLORY

Zdefiniuj paletę kolorów. Przygotowując projekt zdefiniuj podstawową paletę kolorów (tło strony, body text, akcenty) i staraj się nimi operować na przestrzeni projektu. Zbyt duża liczba kolorów i ich odcieni może łatwo i szybko wprowadzić wizualny harmider.

Definiuj nowe kolory zamiast zmieniać opacity. Dobrą praktyką jest definiowanie nowych odcieni i unikanie zmian **opacity/alpha** poszczególnych obiektów lub warstw w celu ootrzymnia nowych kolorów (chyba, że staraymy się uzyskać efekt przezroczystości danego elementu). Ułatwi to pracę developerom, którzy nie muszą sprawdzać ustawień koloru w conajmniej 3 różnych miejscach (kolor obiektu, przezroczystość obiektu, przezroczystość grupy).

Przygotuj zestaw kolorów dla wykresów. Jeśli tworzony przez Ciebie projekt posiada wykresy, zdefiniuj zestaw kolorów (i kolejność w jakich powinny być używane). Staraj się unikać różnych odcieni tego samego koloru, ponieważ obniża to czytelność wykresu (odbiorca z nawet drobną wadą wzroku może mieć spore problemy z ich odczytaniem).

NAGŁÓWKI

Zaprojektuj strukturę nagłówków H1-H6. Poza zdefiniowaniem fontu, rozmiaru i koloru pamiętaj również o takim ustawieniu line height / marginesów, aby dobrze komponowały się z pozostałymi nagłówkami i body text.

FORMULARZE

Kontrolki formularzy (form elements):

- Etykieta (Label)
- Pole tekstowe (Input), Pole numeryczne
- Dropdown (Select)
- Textarea
- File input
- Time picker, Date picker, Date time picker
- Suwak / Zakres (Range)
- Button
- Radio button
- Checkbox / Switch
- Opis pola (np. informacja o minimalnej długości hasła)
- Komunikat błędu
- Informację o obowiązku wypełnienia/możliwości pozostawienia pustym (Required/Optional)

Przygotowanie komponentów pól. Poza samym zaprojektowaniem kontrolek formularzy warto zaprojektować też całe pola (np. złożone z etykiety, pola tekstowego i opisu) tak, aby były elastyczne (można było je rozciągać/zwężać) i zawierały zdefiniowane marginesy/odstęp.

Sekcje formularzy. W przypadku długich/skomplikowanych formularzy, podział ich na sekcje (z nagłówkami i opisami) pozytywnie wpływa na szybkość i skuteczność ich wypełniania.

Układ horyzontalny. Niektóre krótkie formularze (takie jak logowanie w nagłówku strony, czy zapisywanie się do newslettera) można ułożyć poziomo

(pola są wtedy obok siebie).

SZABLONY E-MAIL

Rozważ wykorzystanie gotowych szablonów email. W przypadku e-maili należy mieć świadomość, że ograniczenia różnych klientów pocztowych wyświetlać je nieco inaczej (co wynika z ich ograniczeń) – część z nich radzi sobie z responsywnymi widokami i GIFami, a inne używają uproszczonej wersji Word’a do renderowania treści (serio!).

Przed przystąpieniem do projektowania szablonów e-mail od zera warto rozważyć tylko drobne zmiany (np. kolorów) w szablonach wbudowanych we framework (np. w przypadku Laravela), skorzystanie z gotowych, darmowych szablonów (np. w [Maizzle](#), który pokochają developerzy uwielbiający Tailwind CSS) lub namówienie klienta na wydanie ok. \$20 na [gotowy, komercyjny szablon](#).

Ogólny szablon dla mailingów transakcyjnych. Warto przygotować ogólny widok, który będzie zawierał:

- Logo
- Kolory tła
- Kilka poziomów nagłówków (zwykle H1-H3 wystarczają)
- Body text
- Hiperłącza
- Sposób ulokowania grafik (rozmiar, marginesy)
- Cytat blokowy

Domyślne fonty. Jeśli nasz projekt korzysta z niestandardowych fontów, dla samych e-maili warto wybrać krój, który jest domyślnie dostępny.

Tytuły maili. Sam projekt graficzny i treść maila to nie wszystko – pamiętaj o przygotowaniu tytułu wiadomości.

STRONY STATYCZNE (BOGATE W TEKST)

Przygotuj jeden gotowy szablon strony statycznej, którą będzie traktowana za uniwersalną (do której będziemy wlewali tekst).

Do tego typu stron można zaliczyć regulamin, politykę prywatności, blog czy strony pomocy. Warto przygotować ogólny widok, który będzie zawierał:

- Nagłówki (H1-H6) ze zdefiniowanymi fontami, kolorami, wysokością wierszy i odstępami przed/po. Warto przygotować też wariant obrazujący wygląd gdy nagłówek jest linkiem.
- Body text
- Hiperłącza
- Sposób ulokowania grafik (rozmiar, marginesy)
- Cytat blokowy
- Listy (uporządkowane i nieuporządkowane)
- Podczas projektowania warto pamiętać o przygotowaniu wariantów responsywnych (w zależności od potrzeb i wytycznych projektu)
- Korzystanie z gotowych stylów dostępnych w bibliotekach (Bootstrap, Tailwind CSS)

STRONY Z UWIERZYTELNIANIEM

Do tego typu stron można zaliczyć:

- Rejestrację nowego konta
- Rejestrację z zaproszenia innego użytkownika (w celu dołączenia do konta firmowego)
- Logowanie + wersja dla włączonego uwierzytelniania dwuskładnikowego (2FA)
- Resetowanie hasła
- Usuwanie konta

ONBOARDING

Omów z klientem/PMem potrzebę przygotowania procesu onboarding.

Istniejący i dobrze przygotowany proces onboarding to często kluczowy element, który decyduje o tym czy użytkownik będzie korzystał z aplikacji, czy ją porzuci.

Zdefiniuj cele onboardingu. Cele, które będą chcieli osiągnąć stakeholderzy, będą się różniły w różnych aplikacjach. Skomplikowane aplikacje powinny prowadzić użytkownika za rękę pokazując mu kolejne funkcje, wykorzystać kreatory do wypełnienia pustego workspace’a danymi, etc. (Facebook stawia sobie za cel dodanie 4 osób do znajomych w ciągu pierwszych 20 dni od założenia konta; Slack chce, aby wewnątrz firmy wysłano co najmniej 5000 wiadomości).

HAPPY PATH I UNHAPPY PATH

Projektując ścieżki użytkowników warto zastanowić się co powinno się wydarzyć (i jak powinny wyglądać ekrany), gdy dana akcja się nie powiedzie (np. nie udało się zapisać do newslettera ponieważ e-mail jest niepoprawny, lub znajduje się już na naszej liście).

STRONY BŁĘDÓW

Zrozumiałe komunikaty. O ile Błąd 404 przeszedł już do popkultury, tak inne kody błędów (403: Brak uprawnień (Forbidden), lub 500: Błąd serwera (Internal server error) są niezrozumiałe dla większości użytkowników.

Nie pozostawiaj użytkownika samego. Poza opisaniem problemu w sposób zrozumiały dla wszystkich warto jest dać użytkownikom możliwość kontynuowania (np. “Przejdź do strony głównej”, “Napisz do nas na support@companyname.com, żebyśmy mogli Ci pomóc”) zamiast pozostawiania ich w ślepym zaułku.

Dodatkowe materiały

Poza samymi plikami graficznymi, na projekty składają się również:

ZDJĘCIA I ILUSTRACJE

Licencje i prawa do wykorzystania. Upewnijmy się, że posiadamy należyte licencje, które pozwalają nam na korzystanie z nich w danym polu eksploatacji.

Poza osadzeniem takiej grafiki w projekcie przekazujemy je również jako osobne pliki. Jeśli na grafiki w projekcie nakładaliśmy filtry lub efekty, to reprodukowujemy je na plikach w wysokiej rozdzielczości i dopiero w takiej formie eksportujemy a następnie przekazujemy.

Zdjęcia i ilustracje przekazujemy nieprzycięte, w maksymalnie wysokiej rozdzielczości. Ułatwi to osobie je wdrażającej ich dopasowanie do różnych formatów, rozdzielczości i gęstości (pikseli) ekranów.

Nie oszczędzamy na wadze plików. Na koniec lepiej kompresować niż denerwować się na niską jakość. Na koniec lepiej kompresować niż denerwować się na niską jakość.

Formaty eksportu grafik.

- Zdjęcia przekazujemy jako JPEG (z najmniejszą możliwą kompresją)
- Ilustracje wektorowe i ikony przekazujemy jako SVG lub ewentualnie EPS,
- Grafiki rastrowe przekazujemy jako PNG 24bit non-interlaced (bez przeplotu) z włączonym transparency (przezroczystość/kanal alpha).
- Eksportując grafiki do docelowych rozmiarów warto zastanowić się nad nowożytnymi formatami kompresji: WEBP, HEIC/HEIF, AVIF, APNG (tam gdzie da się je zastosować)

LOGOTYPY

Przekazujemy wszystkie warianty w popularnych formatach. Jeśli otrzymaliśmy logotypy od klienta, upewnijmy się, że są one w formatach z którymi poradzi sobie frontendowiec (łatwiej poradzić sobie z SVG lub EPS niż z AI).

Przekaż księgę znaku. Jeśli posiadamy księgę znaku lub brandbook (które zwykle zawierają informacje o kolorach, obszarach ochronnych i przykłady poprawnego/błédnego użycia) – przekażmy je razem z logotypami.

FONTY

- Upewnijmy się, że posiadamy należyte licencje, które pozwalają nam na korzystanie z nich w danym polu eksploatacji.
- Jeśli licencja lub koszty nie pozwalają nam na wykorzystanie danej rodziny czy kroju w projekcie, poszukajmy darmowego lub tańszego zamiennika na Google Fonts, Font Squirrel, czy Adobe Typekit (płatny)
- W przypadku problematycznych/egzotycznych formatów plików możemy spróbować dokonać konwersji (o ile pozwala nam na to licencja)
- W przypadku projektów dla www starajmy się ograniczyć liczbę rodzin i krojów na poszczególnych stronach – poprawi to czytelność i zredukuje rozmiar strony (co przełoży się na szybkość jej ładowania i wyniki w Lighthouse).
- Jeśli to możliwe ograniczajmy charsety (np. jeśli planujemy użyć jakiś specjalny font monospaced do wyświetlania danych numerycznych, nie będziemy potrzebowali glyphów z cyrylicy)

IKONY

Przygotowując projekt www zadbajmy o faviconę (w dobie rosnących rozdzielczości i [szybko rozwijających się przeglądarek](#) warto zastanowić się nad użyciem SVG) i zestaw ikon (w odpowiednich rozmiarach) dla aplikacji PWA.

[Zbiór dokładnych informacji.](#)

Jeśli w projekcie korzystamy z gotowego zestawu ikon, oprócz jego przekazania (np. w formie paczki plików lub linku do biblioteki dostępnej online) warto nazwać komponenty zawierające ikony tak jak nazwy plików (lub dodawać te nazwy w formie komentarzy) – ogromnie ułatwia to znajdowanie i osadzanie poprawnych ikon.

SOCIAL MEDIA

Nawet jeśli klient nie planuje obecności w mediach społecznościowych (w celach stricte marketingowych), niektóre platformy wymagają założenia kont w celu integracji z ich API.

Open Graph Images. Większość sieci społecznościowych i komunikatorów wyświetla metadane i miniaturę po wklejeniu linku – z tego powodu warto przygotować dedykowane grafiki dla poszczególnych platform (np. Facebook, Twitter). Nie podajemy konkretnych rozdzielczości i formatów, bo te ulegają ciągłym zmianom. Niektóre formaty wymagają odpowiedniego przemyślenia i zaprojektowania, ponieważ ta sama grafika jest przycinana na różne sposoby i wyświetlana na różnych urządzeniach (desktop/mobile).

- <https://www.bannerbear.com/>

Avatary. Część platform społecznościowych operuje awatarami w kształcie kwadratu (zwykle z zaokrąglonymi narożnikami), a część przycina je do kształtu koła. Jeśli niemożliwym jest przygotowanie uniwersalnego obrazka lepiej przygotować dwa osobne. Rozdzielczość **1024×1024** powinna być wystarczająca. Eksportuj je w formie kwadratu, bez przezroczystości (nawet te okrągłe – platformy i aplikacje zawsze same maskują). Dzięki temu nie będzie problemu gdy okrągły avatar (np. z Twittera) zostanie później zamaskowany z użyciem tzw. **Squircle** (jak to miało ostatnio miejsce w Clubhouse, które importowało avatary z Twittera).

POZOSTAŁE

Jeśli wiemy, że przygotowywana strona będzie korzystała z dobrodziejstw PWA, warto przygotować **splash screen** i **ikony aplikacji**.

Wygoda i prędkość narzędzi

Poznaj dokładnie swoje narzędzia. Warto poświęcić trochę czasu na zapoznanie się z dokumentacją, opisami aktualizacji, dostępnymi rozszerzeniami i tutorialami pokazującymi jak korzystać z Figmy, Sketcha i innych programów graficznych. Wiedza tajemna i rękaw pełen sztuczek uczynią z Ciebie jeszcze lepszego magika :)

Obejrzyj najpopularniejsze wtyczki i dodatki. Czasami jeden plugin potrafi zautomatyzować niektóre kroki procesu projektowania i pozwala Ci oszczędzić godzin mordegi.

- Figma: <https://www.figma.com/community>
- Sketch: <https://www.sketch.com/extensions/>
- Oficjalny kanał Figmy na YouTube:
<https://www.youtube.com/channel/UCQsVmSa4X-G3lHlUtejzLA>
- Content Reel Plugin:
<https://www.figma.com/community/plugin/731627216655469013>

Korzystaj ze skrótów klawiszowych. Rób to wszędzie gdzie się tylko da. Niewygodne przemapuj, a brakujące dodaj. Na początku będzie trudno; na końcu nie będziesz się wzdrygać na samą myśl o klikaniu.