Engineer

brukergrensesnitt

den delen av et program en person ser og arbeider med

Hva det er

Et brukergrensesnitt, på engelsk *user interface* og ofte forkortet UI, er skjermene, menyene, feltene, knappene, meldingene og lydene en person bruker for å fortelle et program hva det skal gjøre, og for å finne ut hva det gjorde. Det er også reglene bak dem: hvilken kontroll som kan nås fra hvor, hva som skjer ved et klikk, og hva programmet sier når noe går galt.

Ordlisten til NIST tar det videste synet og definerer et brukergrensesnitt som de fysiske eller logiske midlene brukere samhandler med et system, en enhet eller en prosess gjennom. Den fysiske siden er tastaturet, berøringsskjermen, dreieknappen på et apparat. Den logiske siden er alt som tegnes eller skrives ut som svar: et vindu med menyer, en ledetekst som venter på en skrevet kommando, en stemme som ber om et tall. Et grafisk grensesnitt, en kommandolinje og en talemeny er alle brukergrensesnitt, og ett program har ofte flere av dem.

På nettet gir WCAG byggeklossen et navn. En brukergrensesnittkomponent (*user interface component*) er en del av innholdet som brukerne oppfatter som én kontroll for én bestemt funksjon. Merknadene til definisjonen teller komponenter etter hva brukeren oppfatter som én kontroll, og ikke etter hvordan den er programmert. Ett programmatisk element kan derfor romme flere komponenter. Standardens eget eksempel er en kontroll som flytter seg gjennom innholdet linje for linje, side for side eller til et vilkårlig punkt, der hver måte å flytte seg på trenger sitt eget navn og teller som en egen komponent. Skjemaelementer, lenker og kontroller som skript har bygget, teller alle med.

To nabobegreper følger med. UX, brukeropplevelse (*user experience*), er det videre spørsmålet om hvordan det er å bruke produktet fra start til slutt: om oppgaven blir gjort, hvor lang tid den tar, og hvordan feil blir håndtert.

Brukergrensesnittet er den delen av opplevelsen du kan peke på på en skjerm. Et designsystem er et felles sett med komponenter, designtokens og regler som et team bygger grensesnittet sitt av, slik at to skjermer laget av to personer i to ulike måneder fortsatt ser ut og oppfører seg som ett produkt.

Bruk begrepet i en beslutning

Bruk «Hva er et brukergrensesnitt?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Hvorfor det er viktig

Verdien i et program når fram til folk gjennom grensesnittet, eller den når ikke fram i det hele tatt. En funksjon som bare den som allerede vet hvor den er, finner, er for de fleste brukere aldri levert. Et slikt gap dukker sjelden opp i en testpakke, fordi hver funksjon virker. Det dukker opp i spørsmål til kundestøtten og i utfyllinger som blir avbrutt halvveis.

Konsistens læres én gang og brukes om igjen. Folk tar med seg det de lærte på én skjerm, til den neste. Suksesskriterium 3.2.4 i WCAG 2.2, Consistent Identification, krever at komponenter med samme funksjon innenfor et sett nettsider identifiseres på samme måte. Understanding-dokumentet til kriteriet gir grunnen med enkle ord: når samme funksjon har ulike navn på to steder, kan folk tro at de to gjør forskjellige ting, eller bruke tid på å finne ut at de gjør det samme.

Feil er en del av grensesnittet. Hvert skjema møter ugyldig inndata, og måten det svarer på, er like mye en designbeslutning som oppsettet. Suksesskriterium 3.3.1, Error Identification, krever at en feil som oppdages, navngir feltet og beskriver problemet i tekst. Kriterium 3.3.3, Error Suggestion, går et steg videre: når programmet vet hvordan automatisk oppdagede ugyldige inndata kan rettes, gir det forslag, med mindre det ville sette innholdets sikkerhet eller formål i fare. Eksempelet i Understanding-dokumentet er et felt som venter et månedsnavn, der en person som skriver «12», får tilbud om enten listen over godkjente verdier eller spørsmålet «Mente du desember?».

Hver kontroll må også kunne forklares for programvare. Kriterium 4.1.2, Name, Role, Value, krever at navnet og rollen til hver brukergrensesnittkomponent, også dem skript har bygget, kan bestemmes av programmer, og at endringer i tilstand meldes til dem. En kontroll som bare ser ut som en kontroll på skjermen, er usynlig for hjelpemidlene mange er avhengige av for å nå den.

Slik ser det ut i praksis

Det tydeligste tilfellet er en maritim plattform der produktets UI og UX ble designet fra ende til annen. Den viser fagpersoner, selskaper, skip, stillinger og omtaler, og den måtte være tett nok til å romme ekte maritime data uten å bli en vegg. En håndfull beslutninger bar den. Katalogrutenettene veksler mellom kort og tabell, og appen husker hvilken visning personen valgte. Hvert skjema viser kravene sine over inndatafeltet som en levende liste, der hver regel blir grønn når den er oppfylt og ravgul så lenge den mangler, så folk blir veiledet mens de skriver i stedet for å bli irettesatt etter at de har sendt inn. Brødsmulestier og et tokolonneoppsett går igjen på hver entitetsside, og det får separate skjermer til å henge sammen som ett produkt. Feil fører et sted: en omdirigering i stedet for en blindvei. Siden fører tillitsargumentet rett ut. På en plattform folk bruker til karrieren sin, ville et slurvete utseende få dem til å stole mindre på dataene, og det med rette.

Den kommersielle siden kommer tydeligst fram i 200 forbedringer av UI og UX for en kartapplikasjon med GIS. Grensesnittet hadde blitt kraftig og komplisert, med god funksjonalitet bak klossete interaksjoner. I stedet for å gjette tok arbeidet utgangspunkt i tilbakemeldinger og bruksdata, laget en backlog med 200 konkrete endringer, rangerte dem etter effekt og førte de viktigste gjennom teamet til de ble levert som reelle funksjoner. Grensesnittet ble merkbart enklere å bruke, og arbeidet bidro til at inntektene ble tidoblet.

Å holde et grensesnitt konsistent over tid er et eget problem, og det blir tatt opp i et designsystem i stilen «Liquid Glass» fra iOS 26. Det visuelle språket bruker gjennomskinnelige paneler med bakgrunnsuskarphet, lagdelte skygger, litt lysbrytning og fjærende animasjoner, bygget av verktøyklasser i Tailwind. Hver skjerm settes sammen av et fast register av komponenter, bl.a. Card, Label, Button, GlassIconButton og DirectoryGrid. Lint-regler avviser et hjemmesnekret merke eller en hjemmesnekret brikke og peker mot den kanoniske komponenten, og dokumentasjonen setter konkrete feil ved siden av løsningene på dem. Konsistensen avhenger da ikke lenger av hvem som gjør gjennomgangen den dagen, men holdes oppe av verktøyene.

Hvor det går galt

- **Fargen bærer betydningen alene.** Et felt med grønn kant for gyldig og rød for ugyldig forteller ingenting til den som ikke skiller de to fargene fra hverandre. Understanding-dokumentet for Use of Color nevner akkurat dette tilfellet og ber om et visuelt tegn til. Et ord, et ikon eller en endret form ved siden av fargen gjør jobben.
- **En boks som er tegnet som en kontroll, er bare en boks.** En stylet boks med en klikkhåndterer ser ut som en knapp for den som bruker mus, og finnes ikke for den som bruker tastatur eller skjermleser. Plattformens egne kontroller kommer med rolle, fokus og tastaturoppførsel som en etterligning må bygge opp igjen for hånd, bit for bit.
- **Hver skjerm finner opp komponentene sine på nytt.** Hver ny visning får sine egne knapper, merker og kort, alle litt forskjellige, til én enkelt endring betyr å redigere fem versjoner av det samme. Ett register av komponenter, håndhevet av en linter og ikke av hukommelsen, holder avdriften ute.
- **Feilene kommer sent og langt fra årsaken.** Et skjema som godtar alt og deretter viser en rød blokk øverst med alt som var galt, sender folk på leting etter feltene. Navngi feltet, beskriv problemet med ord rett ved siden av det og foreslå en retting, så blir den samme sjekken til veiledning.
- **Grensesnittet er bygget for pulten til den som laget det.** En stor skjerm, en mus, skarpt syn og en rask forbindelse beskriver personen som laget det, og ikke dem som skal bruke det. En telefon holdt i én hånd, en forstørret side eller et tastatur uten mus avslører hver for seg problemer som en pult aldri viser.

Les mer

- NIST CSRC Glossary: user interface — definisjonen på én linje fra NIST SP 800-152, som dekker både de fysiske og de logiske midlene for å samhandle med et system.
- WCAG 2.2: user interface component — definisjonen ut fra oppfattet kontroll, med merknader om hvordan komponentene forholder seg til elementene som implementerer dem.
- Understanding SC 3.2.4: Consistent Identification — hvorfor komponenter med samme funksjon bør ha samme navn på tvers av et nettsted, og hvilke gjentatte komponenter det betyr mest for.
- Understanding SC 3.3.3: Error Suggestion — hva et skjema bør si når det vet hvordan ugyldig inndata kan rettes, med gjennomarbeidede eksempler.
- Understanding SC 1.4.1: Use of Color — når farge kan bære betydning, og når et visuelt tegn til er påkrevd.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er et brukergrensesnitt?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/ui/