

Revolutioner din testproces med ChatGPT!

Brug ChatGPT som din sparringspartner i dagligdagen

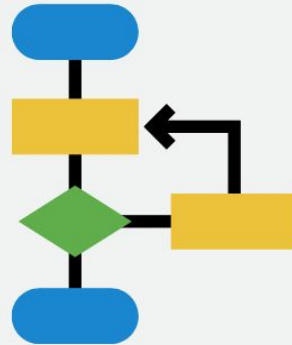
Effektiviser testprocessen gennem dialog med chatGPT

Teststrategier, testanalyse, testdesign, test af kode - mulighederne er mange

Brug af AI bliver en **“gamechanger”** - få inspiration til at arbejde mere effektiv ved brug af ChatGPT

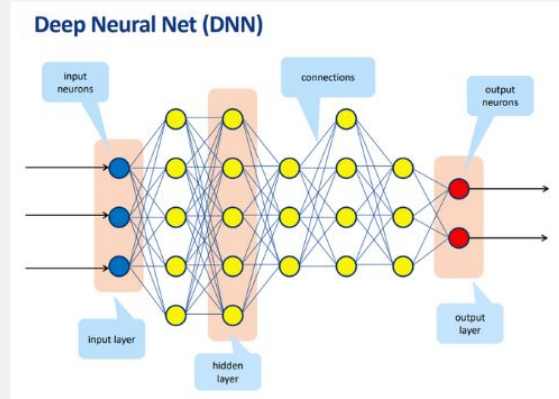


KODET LOGIK

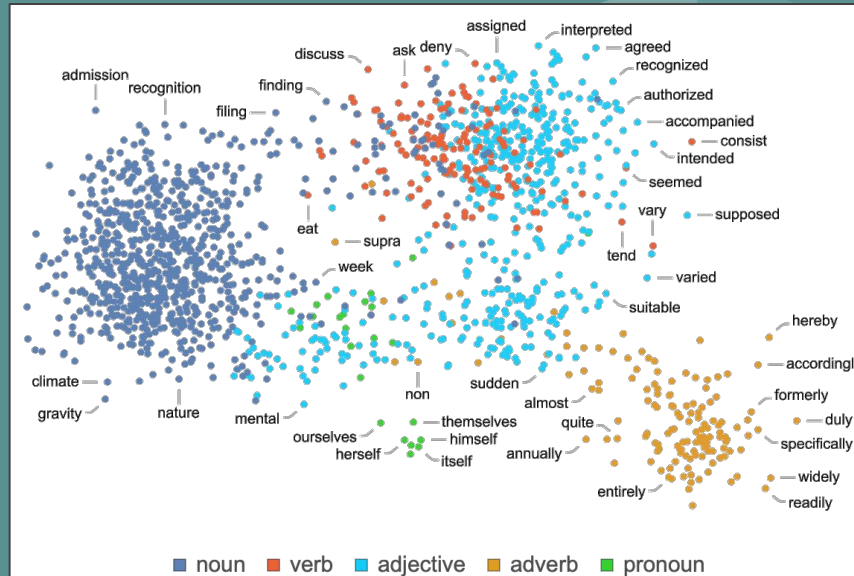
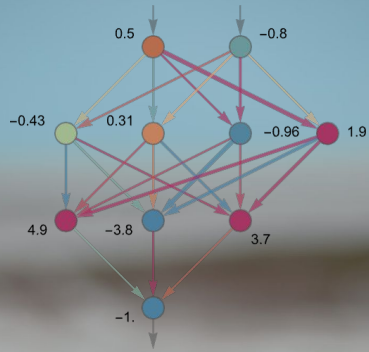


DETERMINISTISK

TRÆNET MODEL



**NON-DETERMINISTISK
TRÆNET TIL AT GIVE "GODE
NOK" SVAR**



[Attention is all you need](#)

[What Is ChatGPT Doing ... and Why Does It Work?](#)



ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

TRY CHATGPT ↗

<https://chat.openai.com/chat>

For denne session antag du er erfaren test specialist.

Foretag testanalyse af følgende og list relevante testscenarier:

Givet et scenarie hvor en fond tildeler penge til projekter i forskellige organisationer. Alle projekter får 10000 DKK.

Hvis et projekt bruger færre penge end tildelt skal overskydende midler betales tilbage når projektet er afsluttet.

Hvis et projekt bruger flere penge end tildelt skal organisationen financiere projektet selv differencen.

Model: GPT-4



For denne session antag du er erfaren test konsulent. Brug dansk som sprog. Foretag testanalyse af følgende og list relevante testscenarier:

Givet et scenarie hvor en fond tildeler penge til projekter i forskellige organisationer. Alle projekter får 10000 DKK.



Hvis et projekt bruger færre penge end tildelt skal overskydende midler betales tilbage når projektet er afsluttet

Hvis et projekt bruger flere penge end tildelt skal organisationen financiere projektet selv differencen.

Lad os lave en testanalyse





Som en erfaren testkonsulent vil jeg foreslå følgende testscenarier for at dække forskellige aspekter af systemet og sikre dets korrekte funktionalitet:  

1. Test af normal tildeling af penge til projekter:
 - a. Opret et projekt og kontroller, at det modtager 10.000 DKK i fondsmidler.
 - b. Opret flere projekter og kontroller, at hver af dem modtager 10.000 DKK i fondsmidler.
2. Test af tilbagebetaling af overskydende midler:
 - a. Afslut et projekt, der har brugt mindre end 10.000 DKK og kontroller, at overskydende midler betales tilbage til fonden.
 - b. Afslut flere projekter med forskellige beløb af ubrugte midler og kontroller, at de korrekte overskydende beløb betales tilbage til fonden for hver.
3. Test af finansiering af projekter, der overstiger tildelte midler:
 - a. Afslut et projekt, der har brugt mere end 10.000 DKK, og kontroller, at organisationen finansierer differencen.
 - b. Afslut flere projekter, der har brugt forskellige beløb over 10.000 DKK, og kontroller, at organisationerne finansierer de respektive differencer.
4. Test af fejlhåndtering og validering:
 - a. Prøv at oprette et projekt uden nødvendige oplysninger og kontroller, at det ikke er muligt.
 - b. Prøv at afslutte et projekt, der ikke eksisterer, og kontroller, at det ikke er muligt.
 - c. Prøv at tildele et negativt beløb til et projekt og kontroller, at det ikke er muligt.
5. Test af rapporter og opfølgning:
 - a. Generer en rapport over alle projekter, der har modtaget fondsmidler, og kontroller, at alle tildelte beløb er korrekte.
 - b. Generer en rapport over alle afsluttede projekter og deres tilbagebetalte beløb og kontroller, at de er korrekte.
 - c. Generer en rapport over alle projekter, hvor organisationerne har finansieret differencen, og kontroller, at de angivne differencer er korrekte.
6. Test af ydeevne og skalering:
 - a. Opret og afslut et stort antal projekter samtidig og kontroller, at systemet fungerer korrekt og uden forsinkelser.
 - b. Skalér antallet af samtidige brugere og kontroller, at systemet fortsat fungerer effektivt.

Svar dækker testscenarier for specifikation

+

hvad man typisk vil teste for en applikation af denne type



7. Test af sikkerhed og adgangskontrol:

- a. Kontroller, at kun autoriserede brugere kan oprette, redigere og afslutte projekter.
- b. Kontroller, at brugernes adgangsniveauer er korrekt implementeret, så de kun har adgang til de funktioner, de har rettigheder til.
- c. Test for sårbarheder som SQL-injektion og cross-site scripting for at sikre, at systemet er sikkert mod angreb.

8. Test af databas og genoprettelse:

- a. Simuler et utilsigtet databas og kontroller, at der er mulighed for at gendanne data fra sikkerhedskopier.
- b. Test systemets tolerance over for hardware- eller netværksfejl og kontroller, at det kan genoprette normal funktion efter sådanne hændelser.

9. Test af integration med eksterne systemer:

- a. Kontroller, at systemet kan integreres med regnskabs- og faktureringsystemer for at lette rapportering og økonomisk styring.
- b. Test integrationen med andre organisationssystemer, såsom projektledelsesværktøjer og kommunikationsplatforme.

10. Test af brugervenlighed og tilgængelighed:

- a. Kontroller, at systemets brugergrænseflade er intuitiv og let at bruge for alle brugere, uanset deres tekniske niveau.
- b. Test systemet for tilgængelighed og sikre, at det kan bruges af personer med forskellige handicap og funktionsnedsættelser.

11. Test af systemets evne til at håndtere forskellige valutaer og internationale projekter:

- a. Kontroller, at systemet kan håndtere projekter i forskellige valutaer og omregne tildelte beløb og tilbagebetalinger korrekt.
- b. Test systemets evne til at håndtere internationale projekter og sikre, at det kan administrere forskellige skatte- og lovgivningsmæssige krav.

12. Test af systemets evne til at håndtere ændringer i projektets varighed og omfang:

- a. Kontroller, at systemet kan håndtere ændringer i projektets varighed og omfang og korrekt justere tildelte midler og tilbagebetalinger baseret på disse ændringer.
- b. Test systemets evne til at spore og rapportere ændringer i projektets varighed og omfang for at lette projektstyring og evaluering.



Svaret er yderst dækkende hvor næste step vil være at frasortere elementer som ikke er relevante for denne løsning, f.eks. forskellige valutaer



List relevante testcases relateret til beløb Brug formen given-when-then. Formater med ny linje efter hvert testscenarie. List kun testcases relateret til midler tildelt et specifikt projekt



1. Tildeling af midler til et projekt:

Givet et nyt projekt er oprettet
Når fonden tildeler midler til projektet
Så modtager projektet 10.000 DKK.

2. Tilbagebetaling af ubrugte midler:

Givet et projekt har brugt 8.000 DKK af de tildelte midler
Når projektet afsluttes
Så betales 2.000 DKK tilbage til fonden.

3. Ingen tilbagebetaling, hvis alle midler er brugt:

Givet et projekt har brugt alle 10.000 DKK af de tildelte midler
Når projektet afsluttes
Så betales der ikke noget tilbage til fonden.

4. Organisationen finansierer meromkostninger:

Givet et projekt har brugt 12.000 DKK
Når projektet afsluttes
Så finansierer organisationen de ekstra 2.000 DKK.

5. Projekt med nul forbrug af midler:

Givet et projekt ikke har brugt nogen af de tildelte midler
Når projektet afsluttes
Så betales alle 10.000 DKK tilbage til fonden.

6. Projekt med delvis forbrug af midler:

Givet et projekt har brugt 6.500 DKK af de tildelte midler
Når projektet afsluttes
Så betales 3.500 DKK tilbage til fonden.

7. Projekt med fuldt forbrug af midler og meromkostninger:

Givet et projekt har brugt 15.000 DKK
Når projektet afsluttes
Så finansierer organisationen de ekstra 5.000 DKK.



Udledte testcases på given-when-then form

Vi afgrænser til at kigge på beløbsscenarioer



List testscenarier i tabelform med angivelse af testcasenr, beløb tildelt, forbrug, difference, beløb der skal betales tilbage, beløb der skal finansieres selv.

Tilføj kolonne med ækvivalens klasse der afspejler relevante ækvivalensklasse for test relateret til beløb.



Testcase nr.	Beløb tildelt	Forbrug	Difference	Beløb der skal betales tilbage	Beløb der skal finansieres selv	Ækvivalensklasse
1	10.000 DKK	0	10.000	10.000	0	Ingen forbrug
2	10.000 DKK	6.500	3.500	3.500	0	Delvis forbrug
3	10.000 DKK	8.000	2.000	2.000	0	Delvis forbrug
4	10.000 DKK	10.000	0	0	0	Fuldt forbrug
5	10.000 DKK	12.000	-2.000	0	2.000	Meromkostninger
6	10.000 DKK	15.000	-5.000	0	5.000	Meromkostninger

I denne tabel er der angivet seks testscenarier med forskellige forbrugsniveauer. Ækvivalensklasserne repræsenterer de forskellige mulige situationer i forhold til beløbene og er opdelt i "Ingen forbrug", "Delvis forbrug", "Fuldt forbrug" og "Meromkostninger". Testscenarierne dækker alle de relevante kombinationer af disse ækvivalensklasser, hvilket sikrer, at systemet håndterer tildeling af midler, tilbagebetaling og selvfinansiering korrekt.

GPT udleder selv regneregler og ækvivalensklasse



reducer testcases til kun en per ækvivalensklasse



Testcase nr.	Beløb tildelt	Forbrug	Difference	Beløb der skal betales tilbage	Beløb der skal finansieres selv	Ækvivalensklasse
1	10.000 DKK	0	10.000	10.000	0	Ingen forbrug
2	10.000 DKK	6.500	3.500	3.500	0	Delvis forbrug
3	10.000 DKK	10.000	0	0	0	Fuldt forbrug
4	10.000 DKK	12.000	-2.000	0	2.000	Meromkostninger

Ved at reducere antallet af testscenarier til en pr. ækvivalensklasse, har vi nu fire testscenarier, der hver repræsenterer en unik situation i forhold til beløbene. Dette vil stadig give en god dækning af de forskellige situationer, systemet skal håndtere, og sikre, at det fungerer korrekt for hver ækvivalensklasse.

Finjusterer testcases - DU skal stadig tænke selv ;-)

Fra GPT3.5 til GPT4 skete der et spring, mindre behov for tilpasning



Generer CSV kode for tabel med testcases. Gør det nemt at kopiere CSV koden

The previous model used in this conversation is unavailable. We've switched you to the latest default model



Sure, her er CSV-koden for tabellen med testcases:

```
css Copy code
Testcase nr.,Beløb tildelt,Forbrug,Difference,Beløb der skal betales tilbage,Beløb
1,10.000 DKK,0,10.000,10.000,0,Ingen forbrug
2,10.000 DKK,6.500,3.500,3.500,0,Delvis forbrug
3,10.000 DKK,10.000,0,0,0,Fuldt forbrug
4,10.000 DKK,12.000,-2.000,0,2.000,Meromkostninger
```

Du kan kopiere ovenstående tekst og indsætte den i en teksteditor eller et regnearksprogram, der understøtter CSV-format.

CSV export til excel

Beriges efterflg med relevante info for testcase i Azure Devops

Den generelle beskrivelse for testcase kan vi bede ChatGPT om at lave



Første prompt

ChatGPT

Uddyb

ChatGPT

Præcisering

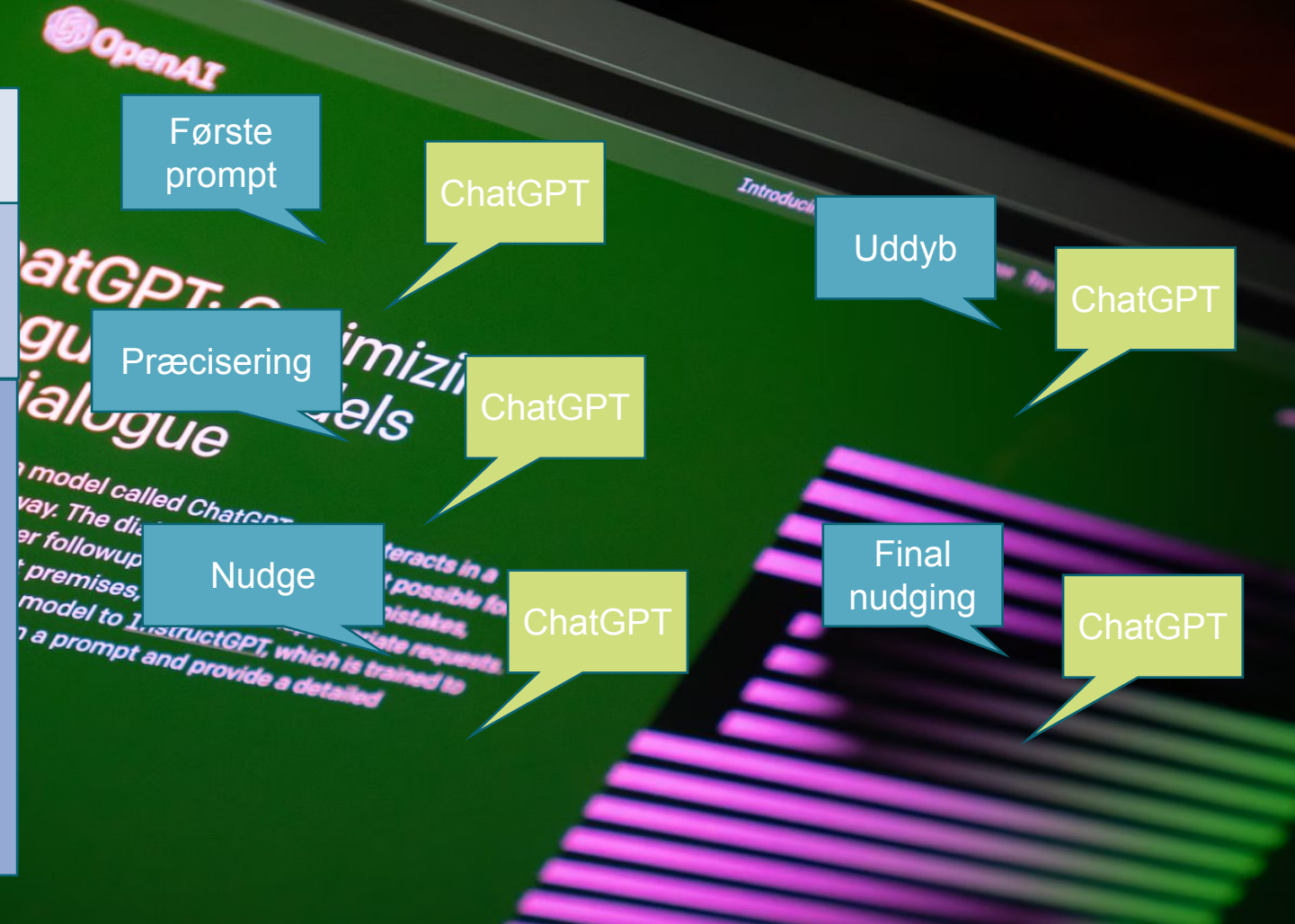
ChatGPT

Nudge

ChatGPT

Final nudging

ChatGPT



Antag du er

Domæne



Antag rolle som erfaren test konsulent med stor viden om test af SAP løsninger.
Foretag test analyse af følgende tekst, list relevante testscenarier med accept kriterier.
Grupper relaterede testscenarier:



Angivelse af hvad der ønskes udført

Analyser Tekst/User Story her udvidet
med acceptkriterier og gruppering



Lav teststrategi for at teste en softwareløsning til en blodbank



For at teste en softwareløsning til en blodbank, kan man bruge følgende teststrategi:

1. Funktionel test: Test at alle funktioner i softwareløsningen fungerer som forventet og er i overensstemmelse med kravene til systemet.
2. Integrationstest: Test at softwareløsningen integrerer sig korrekt med andre systemer, såsom laboratorieudstyr og databaser.
3. Belastningstest: Test hvordan systemet reagerer under høj belastning, for eksempel ved at simulere mange samtidige anmodninger fra brugere.
4. Sikkerhedstest: Test at systemet beskytter patientdata mod uautoriseret adgang og at det overholder gældende lovgivning om databeskyttelse.
5. Brugeracceptttest: Få brugere fra blodbanken til at teste systemet og give feedback om brugervenlighed og funktionalitet.
6. Regressionstest: Test at nye ændringer i systemet ikke påvirker tidligere funktioner.
7. Acceptancetest: Test at systemet opfylder kravene fra kunden og er klar til at blive implementeret i produktion.

ChatGPT kan skitsere punkterne til en teststrategi



List alle testscenarier på formen given-when-then. Grupper relaterede testscenarier under en overskrift. Brug bold for overskrifter. Vær så præcis så mulig for acceptkriterier. Angiv f.eks. den præcise fejltekst der skal tjekkes for hvor det er relevant. Nummerer de enkelte testcases



ChatGPT grupperer relaterede testscenarier med passende overskrift

Given-when-then giver ofte mere præcis resultat



Given the feature description below, list relevant testcases for testing the functionality

Feature: Login into XXXX Application

Scenario Outline: Login using different types of credentials

Given I am in the Login Page

When I try to login with <type> credentials


Then I should be <outcome>

Examples:

```
|type |outcome |
|valid |successfully logged in|
|invalid|shown error message |
```

Hvis beskrivelser er på kendt format, f.eks Gherkin bliver svaret endnu mere “stil sikkert” og uddybende



Relevant test cases for testing the functionality of the "Login into XXXX Application" feature with the scenario outline provided would be: 

1. Test case for valid credentials: Input valid username and password, verify that the user is successfully logged in.
2. Test case for invalid credentials: Input invalid username and password, verify that an error message is displayed.
3. Test case for empty credentials: Input empty username and password, verify that an error message is displayed.
4. Test case for special characters in credentials: Input username and password with special characters, verify that the user is successfully logged in or an error message is displayed as appropriate.
5. Test case for maximum length of credentials: Input maximum allowed length for username and password, verify that the user is successfully logged in or an error message is displayed as appropriate.
6. Test case for minimum length of credentials: Input minimum allowed length for username and password, verify that the user is successfully logged in or an error message is displayed as appropriate.
7. Test case for login using only username: Input only the username and leave the password field empty, verify that an error message is displayed.
8. Test case for login using only password: Input only the password and leave the username field empty, verify that an error message is displayed.



Group related testcases. List groups with related testcases. use bold for the heading



Group 1: Credentials validation test cases



1. Test case for valid credentials
2. Test case for invalid credentials
3. Test case for empty credentials
4. Test case for special characters in credentials
5. Test case for maximum length of credentials
6. Test case for minimum length of credentials
7. Test case for login using only username
8. Test case for login using only password
9. Test case for login with incorrect username but correct password
10. Test case for login with correct username but incorrect password
11. Test case for login with case sensitivity
12. Test case for login with expired password
13. Test case for login with account locked out
14. Test case for login with invalid characters in credentials
15. Test case for login with non-latin characters in credentials

Group 2: Two-factor authentication test cases

15. Test case for login with two-factor authentication enabled

Group 3: Network connection test cases

16. Test case for login with no internet connection

Group 4: URL validation test cases

17. Test case for login with incorrect URL

Group 5: Multiple attempts test cases

18. Test case for login with multiple attempts

Group 6: Role-based test cases

19. Test case for login with different roles

Please note that this is one possible way to group the test cases, but different groups can also be formed based on the needs of your testing.

Første draft hvor jeg ikke har udfrittet chatGPT med supplerende spørgsmål



generate sample cypress code for testcase 1



Here is an example of how Cypress code could be written to test the functionality of the "Login into XXXX Application" feature for the first test case, "Test case for valid credentials":



```
describe('Login into XXXX Application', () => {
  it('Test case for valid credentials', () => {
    cy.visit('http://localhost:3000/login') // navigate to the login page
    cy.get('input[name="username"]').type('john.doe') // input a valid username
    cy.get('input[name="password"]').type('password123') // input a valid
password
    cy.get('button[type="submit"]').click() // click the login button
    cy.get('.message').contains('Successfully logged in') // verify that the
user is successfully logged in
  })
})
```

This code uses Cypress' built-in commands to navigate to the login page, input valid credentials, submit the form, and check that the correct message is displayed.

Please note that this is a simple example, you may need to adjust it according to your application's structure, and also you may need to add more test such as error message validation, etc.

Sample kode for implentering af testcase i Cypress



antag rolle som python udvikler med ekspertise i XML:



Som en python udvikler med ekspertise i XML, vil din primære rolle være at arbejde med XML-dokumenter og relaterede teknologier ved hjælp af Python-programmeringssprog. Dine opgaver kan omfatte:

1. Parsing og generering af XML-dokumenter: Du skal kunne analysere og forstå XML-dokument



Har fået denne fejl. hvad kan problemet være:

```
C:\Users\██████\pip install contextlib
Defaulting to user installation because normal site-packages is not writeable
ERROR: Could not find a version that satisfies the requirement contextlib (from versions:
none)
ERROR: No matching distribution found for contextlib
```



Fejlen opstår, fordi `contextlib` er en del af standardbiblioteket i Python, og derfor behøver du ikke at installere det separat ved hjælp af `pip`.

I stedet for at installere `contextlib`, kan du importere det i dit Python-script ved at skrive følgende linje øverst i din fil:

```
arduino
import contextlib
```

Hvis du stadig har problemer med at bruge `contextlib`, kan det skyldes, at du har en ældre version af Python. `contextlib` blev først introduceret i Python 2.5 og er tilgængelig i alle senere versioner af Python. Sørg for at opdatere din version af Python, hvis det er nødvendigt.

Debug af fejl ved installation af Python runtime miljø

ChatGPT er imponerende god til at debugge - også mere komplekse fejl

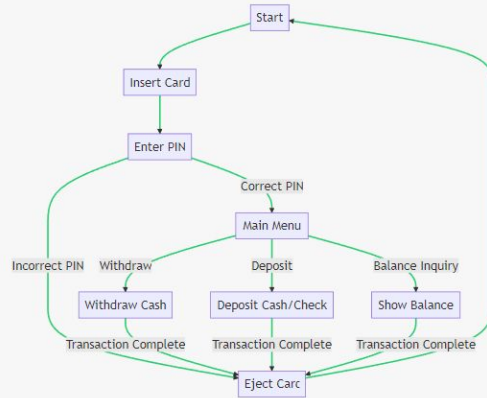


show me a state transition diagram for an atm machine



Used Show Me

Here is the state transition diagram for an ATM machine:



You start by inserting your card into the ATM. After that, you're prompted to enter your PIN. If the PIN is correct, you're taken to the main menu where you can choose to withdraw cash, deposit cash/check, or inquire about your balance. Once the transaction is complete, the card is ejected and the process starts over.

[You can edit this diagram online if you want to make any changes.](#)

If you're interested in contributing to the development of this plugin, you can check out the open source project [here](#).



adjust diagram with following:

- from Main menu an exit option
- from Withdraw, deposit and show balance activity you return to main menu
- if incorrect pin is entered you have two more tries to enter correct pin before card is ejected

ChatGPT4 plugin update

Generating af diagram via **Showme** plugin.
Diagram kode er Mermaid

Alternativ for ChatGPT-3.5

Generate sample mermaid code (in a text box) for XXX

Man vil med ChatGPT-3.5 typisk skulle bruge mere tid på at tilpasse Mermaid kode til at blive "god nok"

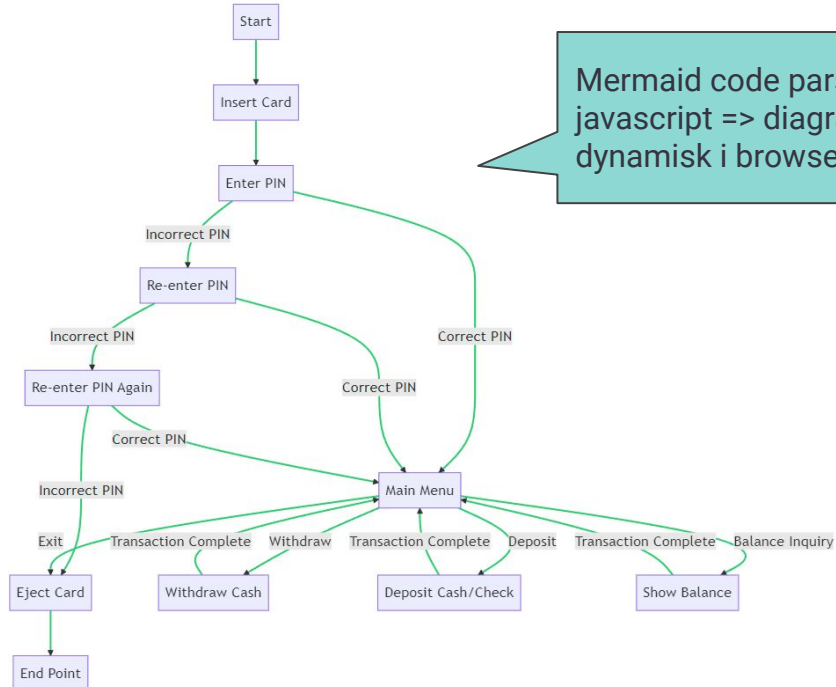
```
1 graph TD
2   S["Start"]
3   S --> I["Insert Card"]
4   I --> P["Enter PIN"]
5   P -- "Correct PIN" --> M["Main Menu"]
6   P -- "Incorrect PIN" --> IP["Re-enter PIN"]
7   IP -- "Incorrect PIN" --> IP2["Re-enter PIN Again"]
8   IP2 -- "Incorrect PIN" --> E["Eject Card"]
9   IP -- "Correct PIN" --> M
10  IP2 -- "Correct PIN" --> M
11  M -- "Withdraw" --> W["Withdraw Cash"]
12  M -- "Deposit" --> D["Deposit Cash/Check"]
13  M -- "Balance Inquiry" --> B["Show Balance"]
14  M -- "Exit" --> EP["End Point"]
15  W -- "Transaction Complete" --> M
16  D -- "Transaction Complete" --> M
17  B -- "Transaction Complete" --> M
18  E --> EP
19  linkStyle 0 stroke:#2ecd71,stroke-width:2px;
20  linkStyle 1 stroke:#2ecd71,stroke-width:2px;
21  linkStyle 2 stroke:#2ecd71,stroke-width:2px;
22  linkStyle 3 stroke:#2ecd71,stroke-width:2px;
23  linkStyle 4 stroke:#2ecd71,stroke-width:2px;
24  linkStyle 5 stroke:#2ecd71,stroke-width:2px;
25  linkStyle 6 stroke:#2ecd71,stroke-width:2px;
26  linkStyle 7 stroke:#2ecd71,stroke-width:2px;
27  linkStyle 8 stroke:#2ecd71,stroke-width:2px;
28  linkStyle 9 stroke:#2ecd71,stroke-width:2px;
29  linkStyle 10 stroke:#2ecd71,stroke-width:2px;
30  linkStyle 11 stroke:#2ecd71,stroke-width:2px;
31  linkStyle 12 stroke:#2ecd71,stroke-width:2px;
32  linkStyle 13 stroke:#2ecd71,stroke-width:2px;
33  linkStyle 14 stroke:#2ecd71,stroke-width:2px;
34  linkStyle 15 stroke:#2ecd71,stroke-width:2px;
35
```

> Sample Diagrams

> History



> Actions



Mermaid code parses as javascript => diagram renders dynamisk i browser

Prompt skabelon

❖ Antag rolle som **Indsæt rolle her**

- Testmanager
- Testanalytiker
- Teknisk tester
- Web udvikler
- SAP Specialist
- ABAP udvikler

med stor erfaring i **indsæt ekspertise eller domæne her**

- Testmanagement
- Testanalyse
- Test af xxx
- SAP modul
- Java/Angular/React/ ABAP/ ...

❖ Analyser nedenstående tekst/user story/kode. Udled/forklar **indsæt ønskede resultat her**

- Testplan for xxx
- Testscenarier med acceptkriterier
- Eksempler på testdata til hvert testscenarie
- Relevante unit test for xxx
- Generer sample unit test kode for xxx anvend **indsæt framework**
- Mermaid diagram der beskriver ...
- ...

Eksempler på supplerende direktiver

- Nummerer de enkelte testscenarier
- Udelad testscenarie xx
- List alle testscenarier på formen given-when-then
- Grupper relaterede testscenarier under en overskrift
- Brug bold for overskrifter
- List og forklar funktionalitet i kode
- Opsummer alle krav/testscenarier
- “Fortsæt, gentag seneste linie” (når svar ikke er komplet)

Prøv dig frem. Husk at man med fordel kan tilpasse seneste prompt og kan navigere frem og tilbage mellem forskellige versioner af prompt

Det er i dialogen med ChatGPT at magien sker

“AI Genie is out of the bottle”

“Muligvis største ændring siden internettet”

“AI vil være en gamechanger for os der arbejder ved en skærm”

“Målbare hurtigere resultater og bedre kvalitet “

“For visse virksomheder og roller vil man tabe uden brug af AI”

Mængden af AI tools stiger dagligt <https://www.futuretools.io/>

Reklame trailer ...

www.twitter.com/hjortholm (ca 50% af tweets er AI relateret)

www.linkedin.com/in/kimhjortholm ChatGPT/AI posts

www.qnation.dk/events AI webinar efter sommerferien

www.qnation.dk/kurser ChatGPT kursus efter sommerferien