

IT 11 PAT (Task 1 – 5)

2022

ANALISE & ONTWERP





# WAT MOET GEDOEN WORD?

Task 1a - Taakdefinisie

Task 1b - Gebruikersvereistes

Task 2 - Databasis ontwerp

Task 3a – Navigasie / Skermvloei (vloeidiagram)

Task 3b - Grafiese Gebruikerskoppelvlak (GGK)

Task 5 - Toevoer met validering, Verwerking, Afvoer (TVA-tabel)

Task 4 - Datawoordeboek

- ▶ Task 4a – Veranderlikes & Komponente
- ▶ Task 4b – Tekslêers
- ▶ Task 4b - Skikkings (*arrays*)

# ONTWERPSDOKUMENT (TASK 1-5):

- ▶ **Skep jou ontwerpdocument in MS Word:**
  - ▶ Skep 'n 'blank new Word Document'
    - ▶ Navigeer na jou Task 1 - folder
    - ▶ Stoor die dokument as **PATontwerp\_VanNaam**.
- ▶ **Verskaf jou word dokument van 'n**
  - ▶ **Voorblad** (*cover page*) met ten minste die volgende inligting:
    - ▶ Die Projeknaam en Logo, PAT ontwerpdocument, jou Naam en Van.
  - ▶ **Inhoudsopgawe** (*table of contents*)
  - ▶ 'n **Header en/of Footer** wat minstens jou **naam en bladsynommers** op elke bladsy sal vertoon.
  - ▶ Maak gebruik van Heading-style om jou **opskrifte** van afdelings (soos in die dokument verder beskryf) te hanteer.

# TASK 1A: TAAKDEFINISIE

Gee 'n beskrywing van:

► jou projek binne die gegewe onderwerp:

**“Tournaments/Competitions for a Sport of your choice”**

*- an interactive tournament style planner and updater*

► die behoefte wat jou program moet aanspreek.

► wat jou program gaan doen om die behoefte wat geïdentifiseer is, aan te spreek.

*LW: GEEN algoritmes & Delphi jargon moet hier gebruik word nie*

**± 150 - 200 WOORDE**



<https://ncert.nic.in/textbook/pdf/kehpl10.pdf> | <https://www.printyourbrackets.com/types-of-tournaments.html>  
<https://slideplayer.com/slide/13690959/> | <https://www.youtube.com/watch?v=pXZV9bKO3h0>

# TASK 1B: GEBRUIKERSVEREISTES (1)

## Tabel vir gebruikerstories (user stories):

Die teikengehoor bepaal die behoeftes en vereistes van die program.

- Identifiseer gebruikers, gebruikersbehoeftes en verwerkingsvereistes van die stelsel (*who, what, why*).
- Jy moet minstens twee akteurs (tipe gebruikers) hê.
- Elke gebruiksgeval kan moontlik 'n vorm/skerm verteenwoordig.
- Lys ten minste 5 take (vereistes) van die program in jou tabel.

## Use Case Diagram vir vloei:

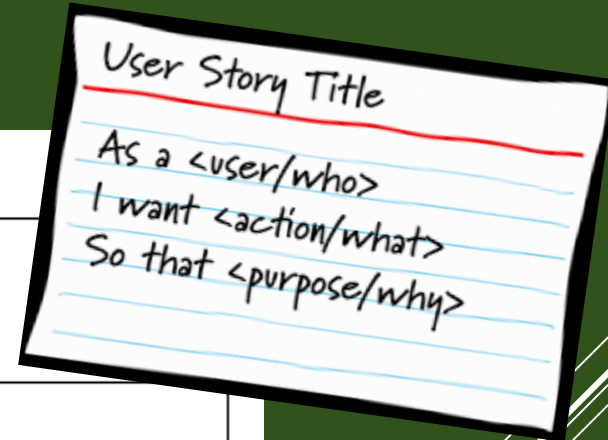
- Teken 'n use case diagram wat die stelselfunksionaliteit (gebruikesgevalle) en **interaksie** tussen die take uiteensit.

# TASK 1B: GEBRUIKERSVEREISTES (2)

## ► Tabel voorbeeld:

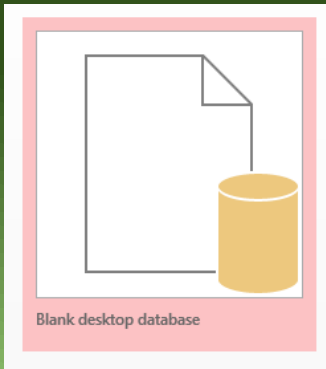
a. **Gebruiker 1:** (Gee aan die gebruiker 'n naam bv. Kliënt/Admin)

<b>Rol</b>	<b>Watter rol vervul die gebruiker in die program?</b> Bv. Hou die data in die databasis op datum.
<b>Aktiwiteite</b>	<b>Wat sal hulle alles doen in the program?</b> Verskaf 'n lys van take wat hierdie gebruiker kan doen bv. Voeg <i>nuwe produk by, verander pryse van produkte, verwyder 'n produk, sien die wins van die besigheid.</i>
<b>Beperkings</b>	<b>Wat kan hierdie gebruiker nie in die program doen nie?</b> Maak 'n lys van dinge wat hierdie gebruiker nie kan doen nie bv. <i>Reset 'n gebruiker se wagwoord, verander die login verslag.</i>



# TASK 2: DATABASIS-ONTWERP (1)

## ► Skep jou Databasis in MS Access



### ► Skep 'n 'blank new database'

- Navigeer na jou Task 2 - folder
- Stoor die databasis met 'n beskrywende naam bv. dbTournament\_XX, waar die XX die eerste letter van jou naam en van voorstel.

### ► Jy moet ten minste twee tabelle hê

### ► Die tabelle moet 'n verwantskap (*relationship*) hê en *referential integrity* afdwing.

# TASK 2: DATABASIS-ONTWERP (2)

## ▶ **Vir elke Tabel (in Access):**

- ▶ Skep die tabel en stoor dit met 'n beskrywende naam bv. *tblSpelers*
- ▶ Ontwerp die hele tabel.
- ▶ Alle velde in die tabel bevat:
  - ▶ Beskrywende name (*wenk: hou dit steeds kort, sonder spasies*)
  - ▶ 'n Beskrywing van die data wat in die veld gestoor word
  - ▶ Gepaste datatipes en veldwydtes



# TASK 2: DATABASIS-ONTWERP (3)

## ▶ **Vir elke Tabel (in jou Word dokument):**

- ▶ Gee 'n gepaste sub-opskrif met elke tabelnaam
- ▶ Gee 'n kort beskrywing van die doel van die tabel.
- ▶ Paste 'n 'screen copy' van die tabel se "design view".
  - ▶ Maak seker dat alle inligting oor die tabel en velde in die Word-dokument beskryf is (*sien vorige slide*).
  - ▶ LW: Die spesifieke datatipes en veldwydtes wys nie in 'design view' nie, voeg dit dus by vir elke veld met 'n voorbeeld van 'n data-item wat in die veld kan wees (of voeg dit in die beskrywing-deel in Access ook by)

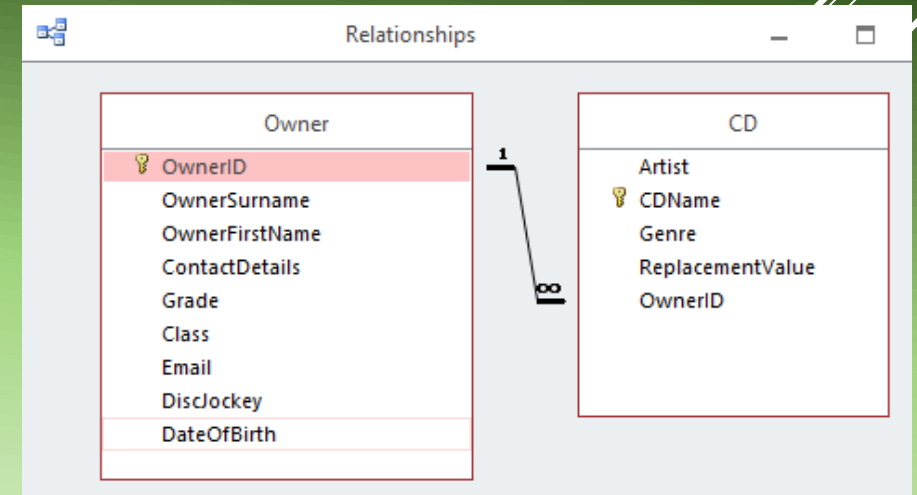
# TASK 2: DATABASIS-ONTWERP (4)

## ➤ Tabel-Verwantskappe (in Access):

- Skep die verwantskap/pe (*relationships*) tussen die tabelle

## ➤ Vir elke tabel-verwantskap (in Word dokument):

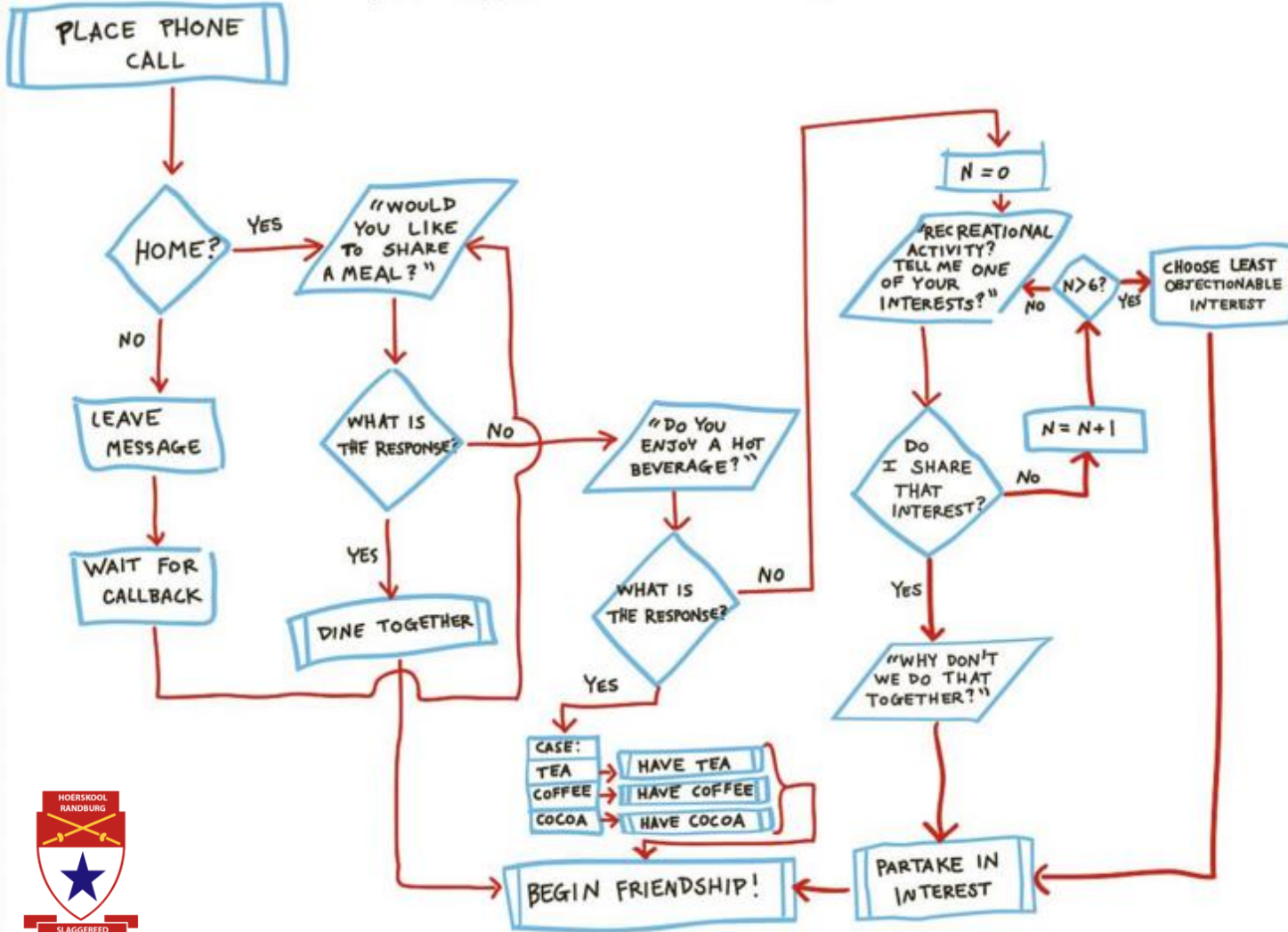
- Gee 'n gepaste sub-opskrif
- Gee 'n kort beskrywing van die verwantskap.
- Paste 'n 'screen copy' uit Access.



# TASK 3A: NAVIGASIE / SKERMVLOEI

## THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, Ph.D



- Maak gebruik van 'n aanlyn-program bv. **DrawIO**.



Untitled Diagram.drawio

File Edit View Arrange Extras Help

- Dui duidelik die logiese programvloei en navigasie tussen tabsheets/skermes aan.
  - Die volgorde van stappe en die vloei van gebeure wanneer die program gaan uitvoer, moet duidelik uitstaan.
- LW die vloeidiagram se doel is om 'n **globale oorsig** van die program aan te bied – dus hoëvlak sonder kode detail.
  - Slegs prosesse en programvloei!

Toevoer  
& Afvoer

Keuses &  
Voorwaardes

Prosesse &  
Take

# PROGRAM-ONTWERP (TASK 3B GGK) (1)

## ► **Skep jou project in Delphi**

- Gaan na (File | Save Project As)
  - Navigeer na jou Task 6 - folder
  - Stoor die Unit (dis die unit van die eerste/main form wat sal vertoon wanneer jou program run) met 'n beskrywende naam bv. frmMain\_XXu, waar die XX die eerste letter van jou naam en van voorstel.
  - Stoor die Project (jy gaan net een project file hê met meer as een unit) met jou stelsel se naam bv. TheTournament\_XXp.

## ► **Jy moet ten minste drie Forms/Tabsheets hê**

# PROGRAM-ONTWERP (TASK 3B GGK) (2)

## ► Vir elke Form (Delphi):

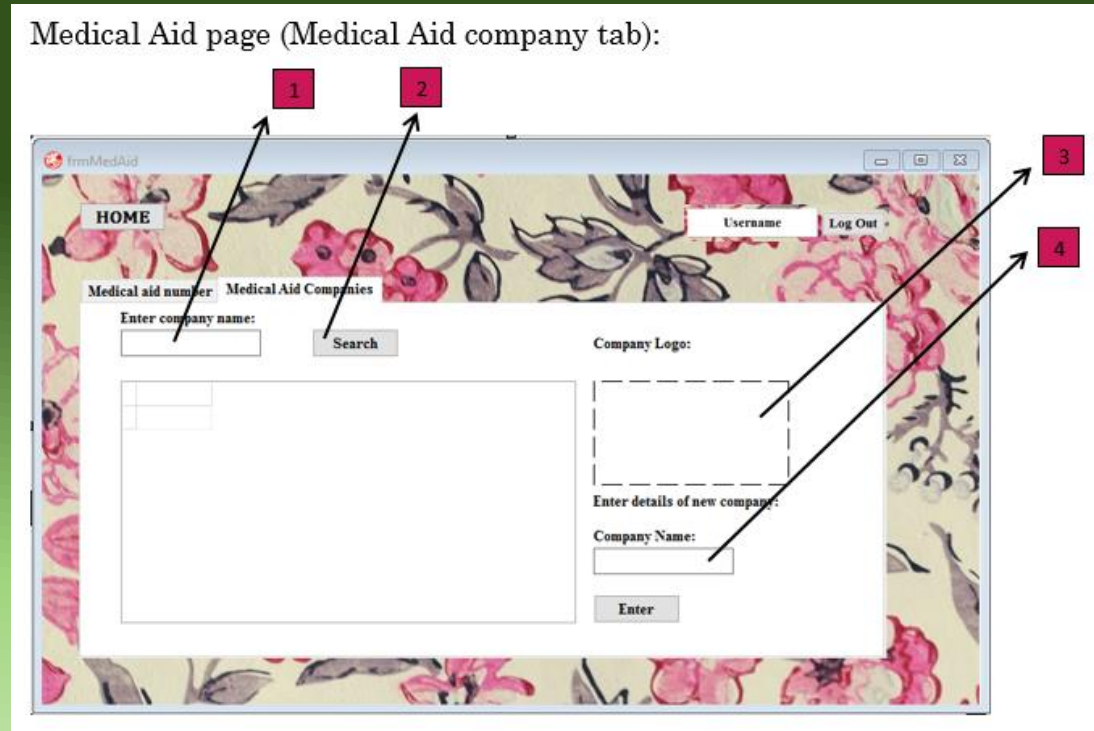
- Skep die addisionele Forms (File | New | Form) in Delphi en stoor dit met 'n beskrywende naam bv. *frmLogin* (File | Save As) in die Task 6 – folder
- Ontwerp die hele form met komponente.  
NB: Hou MRI (HCI)-beginsels in gedagte  
(sien die PAT dokument op p.12)
- Run die program en maak 'n screen shot (alt + PrtScr) van die form wat in jou Word-dokument ge-paste gaan word (of gebruik die Snipping Tool).

HCI principles
(How GUI will meet the users' needs). Does it consider:
<input type="checkbox"/> Purpose of program and user
<input type="checkbox"/> Standard GUI design principles
<input type="checkbox"/> Ease of use, logic flow
<input type="checkbox"/> Clearly marked navigation
<input type="checkbox"/> Friendly dialogue
<input type="checkbox"/> Help

# 5: PROGRAM-ONTWERP (GGK) (3)

## ► Vir elke Form (in Word dokument):

- Gee 'n gepaste sub-opskrif vir elke form
- Gee 'n kort beskrywing van die doel / funksionaliteit van die form.
- Paste die form screen copy (in runtime) onder die sub-opskrif en benoem alle komponente van die form.
- Lys die komponente met hul name – onthou Delphi naming convention!





# 5: PROGRAM-ONTWERP (GGK) (4)

## ► Stel die TVA (IPO)-tabel saam vir die form:

Beskryf alle toevoer, verwerking en afvoer in tabelvorm

### ► Toevoer:

- Beskryf die toevoerbron, datatipe, formaat en toevoercomponent

- Toevoervalidasie:

Beskryf die validering van minstens **drie** verskillende datatipes.

Beskryf die validering van minstens **drie** toevoere.

Dui aan watter foutboodskappe met die validering geassosieer word.

### ► Verwerking:

- Beskryf ten minste **ses** prosesse (what)

- Verskaf algoritmes/formules vir ten minste **vier** prosesse (how)

### ► Afvoer:

- Beskryf die data, formaat en afvoercomponent

# 5: PROGRAM-ONTWERP (GGK) (5)

Input: Keyboard & mouse	Processing	Output
<b>Enter company name into edit (1)</b>  <b>Click on 'Search' button</b>	Company name → [string]  Validation: <ul style="list-style-type: none"><li>• Edit ≠ empty string</li><li>• Data-type validation → n/a</li></ul> Search for matching company name in database  If Company name located in database	         <b>Displays company details → DBgrid</b> <b>Displays company logo</b>

## Voorbeeld van foutafhandeling

'n Geskikte foutboodskap sal vertoon word om foute uit te lig aan die gebruiker  
bv. Geen waarde is by naam ingevul nie:

Johansmedichoicerwp

Geen waarde is by NAAM ingevul nie, asseblief vul 'n waarde in.

OK

## Email(d)→Validation:

- Contains→ @
- Contains→ digits before @ and after @
- Contains→ . (full stop)



# TASK 4: DATAWOORDEBOEK:

## ► Task 4a: Veranderlikes & Komponente

- Hierdie task is reeds in jou IPO tables uiteengesit.

## ► Task 4b: Tekslêers

- Beskryf die relevante gebruik van tekslêer(s)

## ► Task 4c: Skikking/Arrays

- Beskryf 'n goeie toepassing van 'n skikking/s (*array*) wat in jou program gebruik gaan word.

### Variables and components

- ☐ Variety of and correct use of appropriate variable types and components
- ☐ Correct use of local and global variables
- ☐ Proper naming convention of variables for example iNumber, sName etc.
- ☐ Correct prefix for components for example edt, red, cmb etc.

*Let wel: AL hierdie strukture moet sinvol gebruik word en nie data/verwerking uit jou DB dupliseer nie.*

# HOE BENADER EK DIE DOKUMENT?

## ► **My voorstel:**

- Begin met die databasistabelle.
- Skep dan jou GUI met TVA-tabelle
- Voltooi laaste jou gebruikersgevalle, vloedigram en datawoordeboek.

**HET JY AL  
BEGIN????**

**NB:** Maak seker dat jou databasis verskillende tipes velde bevat, en genoeg rekords het (sodat jy AL die punte vir die databasis-afdeling - sien merkblad task 7 – kan kry).