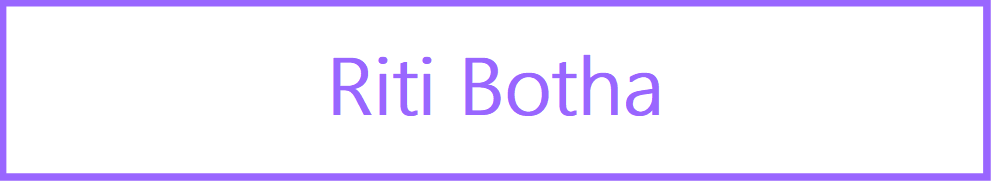
**

# TAAK 1 A – Vereistes Van Die Gebruiker

## Onderwerp:

Die PAT is ‘n assesseeringstaak waar ek ‘n besigheids bestuur sagteware program gaan programmer en so my programmerings en sagteware ontwikkelings vaardighede kan demonstreer. Ek gaan rekenaardenke gebruik om my probleem op te los. Die dokument is die beplanning vir my sagteware – dit sluit die navorsing en probleemdefinisie in. **Ek gaan ‘n besigheids bestuurs sagteware program ontwikkel wat ‘n besigheid (Royale High) gaan bestuur deur markontleding by my sagteware in om te help om ingeligte besluite te neem wat die eienaars van Royale High se inkomste en spelerbetrokkenheid kan beïnvloed.**

## Taak behoefte:

**Die Royale High eienaars het tans nie ‘n manier om te weet watter virtuele items goed om wanneer te verkoop nie en hoeveel mense die virtuele items besit nie.** Die eienaars verloor dus tans ‘n groot aantal geld omdat hulle nie die korrekte besigheids bestuur besluit t.o.v. voorraadbeheer maak nie. Die einddoel van die program is om die eienaars te help om datagedrewe besluite te neem deur insig en aanbevelings aan die Royale High eienaars te gee wat hulle in staat stel **om hul in-speletjie-ekonomieë te optimaliseer** en **inkomste te maksimeer** terwyl 'n **positiewe spelerervaring behou word**.

Figure 1: Die speler dra ‘n fees-gerigte item - die kersfees hoed.

## Hoe my program die probleem sal oplos:

My besigheids bestuurs program gaan markpatrone analiseer om te verseker dat die inkomste wat die einaars moontlik kan verdien gemaksimaliseer word. Ek gaan kyk watter seisoen dit is in Britanje (die meeste spelers is van daar af) en dan gaan ek ook kyk na enige groot feeste en vakansie dae wat opkom en so besluit watter items verkoop moet word. Ek gaan voorstelle gee aan die eienaars soos vir die bekendstelling van nuwe virtuele items, aanpassing van pryse, of aftrede van onderpresterende items.

## Reikwydte:

My program gaan oopbron data gebruik om aanbevelings te maak. My program gaan nie self die items binne Royale High se pryse aanpas of items lewendig kan weg haal of nuut skep nie. My program gaan ook nie speler interaksie lewendig opneem nie en gaan dus nie kan monitor of spelers kroek in die speletjie nie. My program gaan net aanbevelings en informasie gee oor virtuele items en die verkope daarvan. My program gaan ook nie as ‘n koppelvlak tussen spelers en die eienaars dien nie, NET die eienaars sal toegang hê tot my besigheids bestuur sagteware.

# TAAK 1 B – Gebruikers Verhaal en Aanvaardingstoets

## Gebruikersverhaal:

|  |  |  |
| --- | --- | --- |
| ***WIE? (As ‘n...)*** | ***WAT? (Wil ek graag...)*** | ***HOEKOM (Sodat...)*** |
| Besigheidseienaar of bestuurder | * Toegang verkry tot markanalise en aanbevelings deur die toepassing. * Die geskatte wins sien wat ek moontlik kan verdien en inligting kry oor watter items belangrik is om nou uit te saai. * Kyk of Royale High se wins gemaksimaliseer is en so besuilte neem oor hoe om dit te maksimaliseer. | Ek ingeligte besluite kan neem oor die spelekonomie van Royale High en ons inkomste kan maksimaliseer terwyl ek 'n positiewe spelerervaring verseker. |
| Sagteware ingenieurs van Royale High | * Toegang tot inligting van die items hê. * Kyk hoe om te programmeer en verseker dat Royale High op die regte manier en tyd die regte items teen die regte pryse verkoop. * Seker maak dat die eienaars die beste besluite neem en nie menslike foute maak tydens die besluitneming van die proses nie. | Ek kan verseker dat Royale High korrek funksioneer, die nodige aanpassings aan inspeletjie-items maak. |
| Ondersteunings- of kliëntedienspersoneel | **Lyste kry van items wat tans verkoop word en meer inligitng soos die prys of seisoen van die item** om spelernavrae of bekommernisse wat met items of pryse verband hou reg te stel. Hulle sal die toepassing gebruik om vinnig inligting te kry om die klieënte se probleme op te los. | Ek akkurate inligting en hulp aan spelers kan gee, wat spelers satisfaksie verhoog. |

## Aanvaardingstoets:

1. Ek moet gebruik maak van die huidige datum en fees dae wat opkom om te analiseer wat die beste besluit sal wees t.o.v. watter produkte om in ‘n spesifieke tydperk te verkoop teen ‘n sekere prys en ‘n boodskap as uitvoer verskaf aan die gebruiker wat vir my die inligting gaan gee. Wat die aanbeveling sal wees aan die Eienaar en sagteware ingenieurs oor watter items om te verkoop. Hierdie proses gaan van komplekse algoritmes gebruik maak om ‘n persentasie te kry van hoe goeie besluit dit sal wees om elke item op die dag te verkoop, die hoogste paar persentasies sal dan getoon word.
2. Ek moet ‘n lys van produkte (en meer inligting saam met die lyste soos die pryse) wat op die oomblik verkoop word verskaf en ‘n lys van wanneer laas produkte uitgestal is om moontlik verkoop te word kan gee, sodat ons vinnig toegang daartoe kan kry.
3. Die geskatte regtewêreld wins van die binne-speltjie se ekonomie bereken en vertoon as uitvoer. Dit moet die skaarsheid van die items en die pryse daarva in ag neem om die wins te bereken.
4. Ek moet gebruikers in staat stel om inligting oor spesifieke items op te soek deur die itemnaam in te voer. Dit moet inligting soos prys, beskikbaarheid en prestasiegeskiedenis van daardie item vertoon. Die dra by tot die probleemoplossings van probleme wat spelers ervaar.
5. Ek moet deeglike inligting en afvoer oor die in-spel ekonomie en verkope kan genereer. Die rapporte moet gestoor word vir latere verwysing en analise. Die eienaars sal deur hierdie rapporte kan sien of die besigheid se inkomste verbeter en of die regte besluite geneem is. Die sagteware ingenieurs sal deur hierdie rapporte kan sien of Royale High se verkoopsalgoritmes reg funksioneer.
6. Daar moet ‘n inteken blad wees waar net mense met ‘n geldige wagwoord kan inteken om die finansies van Royale High ‘n in-besigheidsgeheim te hou.
7. Nuwe items moet ingevoer kan word. (MOET NIE OU ITEMS VERWYDER NIE, WANT SPELERS HET HIERDIE ITEMS AL REEDS BESIT IN DIE VERLEDE)
8. Die inligting van reeds bestaande items moet verander kan word.
9. Kyk na watter seisoen dit is.



# TAAK 2 – Datawoordeboek

## Veranderlikes en Komponente

**Form 1 - Tabsheet 1: tbsSignIn**

|  |  |
| --- | --- |
| **Veranderlike name en datatipes of**  **Komponent name** | **Beskrywings van die Komponente en Veranderlikes:** |
| btnSignIn  Tbutton  (Button) | Knoppie wat die gebruiker druk na hy sy inteken besonderhede ingetik het, dan sal ‘n algoritme deur ‘n gebruikergedefinieerde metode seker maak dat dit reg in getik is en as als reg is, gaan die gebruiker na tbsData gevat word. |
| imgLogo  Timage  (Image) | ‘n Prent wat altyd “visible” is en die “stretch” funksie altyd true is. |
| edtName  Tedit  (Edit box) | ‘n Edit boks waar die gebruiker hul naam intik, as die naam net uit letters bestaan, is die naam geldig, as dit ongeldig is, verskyn daar ‘n boodskap in lblNameInfo. |
| sNaam  String  Global variable | Stoor die gebruikersnaam waarmee die gebruiker ingeteken het sodat ons later dit weer kan gebruik in die toep. |
| edtPin  Tedit  (Edit box) | ‘n Edit boks waar die gebruiker hul geheime kode intik, as die pin net uit nommers bestaan, is die pin geldig, as dit ongeldig is, verskyn daar ‘n boodskap in lblPinInfo. |
| sPin  String  Local variable | Vier-syfer-pin wat gebruikers toelaat om in te teken. (Kan met 0 begin) |
| **//Meer Inligting onder datastrukture**  arrPin  Array Of string  Local variable | Array van strings wat alle pins bevat in ‘n spesifieke volgorde. |
| **//Meer Inligting onder datastrukture**  arrNames  Array Of string  Local variable | Array van strings wat alle gebruikersname bevat in ‘n spesifieke volgorde.  (Gebruik om intekeningsproses te valideer en word ingelees vanaf NamesAndPins.txt) |
| lblNameInfo  Tlabel  (Label) | Hier verskyn ‘n boodskap oor die naam wat ingetik is indien dit verkeerd ingetik is. – Nie net uit letters nie. |
| lblPinInfo  Tlabel  (Label) | Hier verskyn ‘n boodskap oor die pin wat ingetik is indien dit verkeerd ingetik is. – Nie net uit getalle nie. |
| bfout  (Boolean)  Local variable | As hierdie boolean waarde True is, kan die gebruiker nie inteken nie, as dit false is, kan. |
| pnlHelp  Tpanel  (Panel) | ‘n Self-geskepte foto pop op wat hulp aan die gebruiker gee. **(Dynamic component)** |

**Form 1 - Tabsheet 2: tbsData**

|  |  |
| --- | --- |
| **Veranderlike name en datatipes of**  **Komponent name** | **Beskrywings van die Komponente en Veranderlikes:** |
| dbgItems  Dbgrid  (Database Grid) | Vertoon alle items, vertoon ook later alle gesorteerde items wat vantuit my databasis gekry word. |
| lblWelkom  Tlabel  (Label) | ‘n Label wat ‘n welkom boodskap sal vertoon met die gebruikersnaam. (sNaam) |
| rPersentasie  Real/Float  Local Variable | Die persentasie uitgewerk wat ‘n item in voorraad moet bly/wees.  **// Meer oor die algoritme op die volgende bladsy.** |
| sBoodskap  String  Local Variable | Gee terugvoer oor besigheidsbesluite wat geneem moet word.  Werk uit of item verkoop moet word as item binne seisoen is of moet ophou, buite seisoen. Vir hoeveel diamate item verkoop moet word, as die item verkoop moet word, deur na die skaarsheid van ‘n item te kyk. As die item se skaarsheid 5 is (5 van die top spelers besit die item), word die item steeds verkoop. Onder 5, verkoop, bo 5, nie meer verkoop nie.  As die top 10 spelers 50% van alle items besit, moet ‘n nuwe item uitgesaai word.  Soos in teksleêr (Navrae.txt) in gelees: |
| btnSorteer  Tbutton  (Button) | Sorteer die dbgItems volgens toenemende pryse en vertoon dan op dbgItems |
| btnSoek  Tbutton  (Button) | Soek en vertoon inligting in pnlNavraag oor ‘n spesifieke item. |
| btnNavraag  Tbutton  (Button) | Werk uit watter items verkoop of pryse moet verander word en dan toon sBoodskap op pnlNavraag. |
| pnlNavraag  Tpanel  (Panel) | Vertoon sBoodskap sodra btnNavraag geskep word. |

## DATASTRUKTURE

**Form 1 – Tabsheet 2: tbsData**

|  |  |
| --- | --- |
| **Textfile en skikking verklarings** | **Waarvoor die datastrukture gebruik gaan word** |
| **Array**  ArrNames (of strings)  (Local variable) | Stoor die name wat verkry word uit NamesAndPins.txt om te toets of die gebruiker na tbsData toe toegang het en dat die naam net uit letters bestaan. |
| **Array**  ArrPins (of strings)  (Local variable) | Stoor die pins wat verkry word uit NamesAndPins.txt om te toets of die gebruiker na tbsData toe toegang het en dat die pin net uit syfers bestaan. |
| **Array** ArrLevels (of integers)  (Gloabal variable) | Stoor alle “Levels” van spelers gelees vanuit Leaderboard.txt, aangesien hul name nie aan ons ‘n nut dien nie, maar die hoogste en laagste level van die 10 word in my komplekse algoritmes gebruik om ‘n betaalbare prys te bereken vir sekere items. |
| **Textfile**  NamesAndPins.txt | Stoor alle gebruiker name en pins, geskei met ‘n “;”, sodat dit maklik toetsbaar is en in Microsoft Access en Microsoft Word tabelle gelaai kan word sou dit nodig wees. Die teksleêr word gebruik om die waardes te stoor en sodra die program oop gemaak word, in arrNames en in arrPins in te lees.    Ek gebruik ‘n textfile omdat net hierdie 4 mense toegang nodig het tot my sagteware, so ‘n databasis se RAM-verbruik sal totaal en al onnodig wees. |
| **Textfile**  Leaderboard.txt | Stoor die top 10 spelers se name en “levels”, geskei met ‘n “;”, sodat dit maklik toetsbaar is en in Microsoft Access en Microsoft Word kolomme gelaai kan word. Die Array word gebruik om die levels te stoor en sodra die program begin in arrLevels in te lees. Die rang word in arrLevels in gelees en die hoogste waarde word met die aantal dae tussen vandag en die naaste feesdag geplus en daardie as die teller gebruik wanneer die waarde gedeel word deur 2000 + 20. 2000 Verteenwoordig ‘n aanvaarbare hoogste rang en 20, ‘n gemiddelde aantal dae voor mense ‘n feesdag begin vier. -> Word in rPersentasie is geskryf. |
| **Textfile**  Navrae.txt | Stoor alle navrae se boodskappe wat deur my program geskep is. Die teksleêr dien as ‘n rugsteun vir idees vir die eienaars en word gebruik om groei te sien in die besigheid.  (Werk uit of item verkoop moet word as item binne seisoen is of moet ophou, buite seisoen. Vir hoeveel diamate item verkoop moet word, as die item verkoop moet word, deur na die skaarsheid van ‘n item te kyk. As die item se skaarsheid 5 is (5 van die top spelers besit die item), word die item steeds verkoop. Onder 5, verkoop, bo 5, nie meer verkoop nie.  As die top 10 spelers 50% van alle items besit, moet ‘n nuwe item uitgesaai word.) |
| **User Defined Method**  Function bPrime | Na statistiek deur data analiste ontleed is, het die eienaars van Royale High 'n patroon raak gesien dat produkte wat se heelgetal gedeelte van die prys 'n priemgetal is, beter verkoop word.  Die pryse lyk meer oorspronklik/aanloklik/uniek aan die kopers  Hier toets ek of 'n getal 'n priemgetal is sodat die eienaars vooraf kan bepaal of die produk goed gaan verkoop. |

# Taak 3 – Databasis

## DATABASIS BASIESE ONTWERP

|  |  |  |
| --- | --- | --- |
| **Access Databasis naam: RoyaleHighData.accdb** | | |
| **Tabel 1 naam: tblItems** | | |
| **Die rol/doel van die tabel: Om die items en pryse te sien en te manipuleer. Alle veranderinge word direk in die databasis gestoor.** | | |
| **Veld naam** | **Data tipe en grootte** | **Beskrywing van wat in die veld gestoor sal word** |
| Item | Short text  40 | Items wat elke speler moontlik besit, sodat ons ‘n spesifieke gemiddeld geld kan uitwerk per speler. Kan nie nul waarde hê nie. |
| Prys | Currency | Die aantal diamante wat die ddurste\_item kos. Kan nie nul waarde hê nie. |
| ItemSeisoen | Date/Time  Short Date | Die spesifieke seisoen of vakansiedag waarmee die item verband hou. Kan nul waarde hê. |

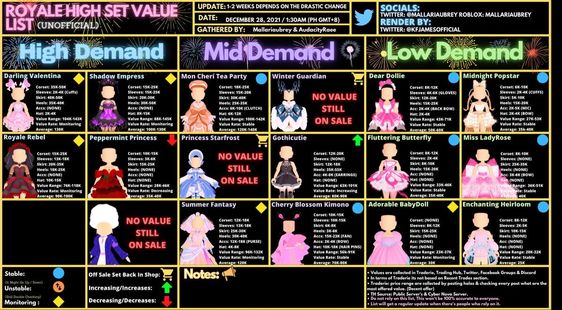
|  |  |
| --- | --- |
| **Manipulasie van tabel:** | |
| **Voeg data by tot die tabel (Individuele rekords)** | Enige gebruiker gaan data by die tabel kan byvoeg. Die gebruiker sal al die velde self moet voltooi. Velde kan nie gelyk wees aan nul nie. Die gebruiker sleutel ‘n datum in ‘n inputbox in vir ItemSeisoen en daardie waarde word dan na ‘n datum veranderlike verwerk. Indien daar nuwe items in die speletjie geplaas word, kan die gebruikers van my sagteware dit maklik invoeg in my sagteware in. |
| **Verander(edit) data in die tabel**  **(Individuele rekords)** | Enige gebruiker gaan die data kan verander, die Prys en ItemSeisoen velde se data kan verander word, maar nie Item nie, aangesien dit konstant gehou moet word vir verwysingsdoeleindes en omdat dit ‘n naam is. Wanneer pryse verander word of spelfoute/tikfoute gemaak is, kan dit herstel word. |
| **Verwyder(delete) data van die tabel.**  **(Individuele rekords)** | Kan nie hier rekords verwyder nie. Geen velde se waardes kan ook gelyk aan nul wees nie. Dit sal die data onakkuraat maak as al die items wat al ooit verkoop was nie in die data vertoon word nie. |
| **Navrae wat geskep gaan word** | 1. Onttrek alle items van ‘n sekere seisoen.   Ek wil alle items uit 'n spesifieke seisoen in die databasis haal. Ek sal alle items soek wat aan die spesifieke seisoen gekoppel is en dit in 'n grid weer gee.   1. Onttrek die duurste 3 items.   Ek sal die items in 'n aflopende volgorde van prys moet sorteer en dan die top 3 items kies.   1. Wys alle items met ‘n spesifieke letter of woord in die naam.   Dit kan gedoen word deur die items wat aan die kriteria voldoen, ophaal, en dit in 'n grid te vertoon. |
| **Verslae wat geskep gaan word** | 1. Gee ‘n lys van alle items **->** Alle Item waardes sal vertoon word in die dbGrid. 2. Gee alle Pryse **->** Alle Prys waardes sal vertoon word in die dbGrid. 3. Gee ‘n lys van alle seisoene **->** Alle Seisoen waardes sal vertoon word in die dbGrid. |

|  |  |  |
| --- | --- | --- |
| **Tabel 2 naam: tblRanglys** | | |
| **Die rol/doel van die tabel: Om uit te werk wat ‘n goeie bedrag sal wees om items teen te verkoop deur na die duurste items en rang (level) van die spelers te kyk. Alle veradneringe word direk in die databasis gestoor.** | | |
| **Veld naam** | **Data tipe en grootte** | **Beskrywing van wat in die veld gestoor sal word** |
| Rang | Number  Integer | ‘n Getal wat die spesifieke speler se rang in die speletjie aandui sodat ons ‘n spesifieke gemiddeld geld kan uitwerk per speler. |
| SpelerNaam | Short text  20 | Die unieke naam van elke speler. (Primêre sleutel)  (Verseker dat elke rekord uniek is) |
| DuursteItem | Short text  40 | Die duurste item wat elke unieke speler besit, sodat ons ‘n spesifieke gemiddeld geld kan uitwerk per speler. |

|  |  |
| --- | --- |
| **Manipulasie van tabel:** | |
| **Voeg data by tot die tabel** | Enige gebruiker gaan data by die tabel kan byvoeg. Die gebruiker sal al die velde self moet voltooi indien die gebruikers die gemiddelde hoeveelheid geld per speler meer akkuraat wil maak deur meer data in te voer. Die gemiddeld sal dal weer uitgewerk word. Maar net die eerste 10 rekords word gebruik met alle ander algoritmes. Die ran, spelernaam en DuursteItem velde sal bygevoeg kan word, DuursteItem waarde sal uit ‘n combobox geselekteer word. |
| **Verander(edit) data in die tabel** | Enige gebruiker gaan die data kan verander, maar slegs die rang en duuurste\_item velde se waardes sal verander kan word, omdat spelernaam konstant gehou moet word vir verwysings doeleindes en omdat dit die primêre sleutel is. Wanneer ‘n speler ‘n hoër rang verdien kan dit verander word of as die speler ‘n nuwe duurste item verkry. |
| **Verwyder(delete) data van die tabel.** | Net die eienaar kan waardes verywder, maar slegs hele rekords kan verwyder word, daar kan nie velde wees wat gelyk is aan nul nie. Aangesien jy nie geen items moontlik kan hê in die speletjie nie en nie kan speel sonder ‘n naam nie en jy altyd ‘n rang groter of gelyk aan 1 sal hê. EN as jy ‘n rekord verwyder, moet ‘n nuwe een ingesit word. Wanneer spelers geblok word van Royale High as gevolg van kroek of oneerlike spel, kan hulle verwyder word van die databasis af. |
| **Navrae wat geskep gaan word** | 1. Onttrek die speler met die hoogste rang.   Dit kan gedoen word deur die spelers in die databasis te sorteer volgens hul rang en dan die speler met die hoogste rang te kies.   1. Onttrek die spelers met die 5 duurste items.   Om dit te doen, sal ek die spelers se items in 'n aflopende volgorde van prys sorteer en die top 5 spelers in die dbgrid vertoon.   1. Wys die rang en duurste item van ‘n spesifieke speler met ‘n spesifieke naam of letter in naam.   Ek wil 'n spesifieke speler vind deur te soek na hul naam of 'n spesifieke letter in hul naam, en dan die speler se rang en duurste item toon. |
| **Verslae wat geskep gaan word** | 1. Gee ‘n lys van alle range **->** Ek wil 'n lys van alle rang in die databasis verkry en dit in die grid gee. 2. Gee alle speler name **->** Alle spelername sal vertoon word in die dbgrid. 3. Gee alle items wat die duurste items is van verskeie spelers **->** Alle DuursteItem waardes sal vertoon word in die dbGrid. |

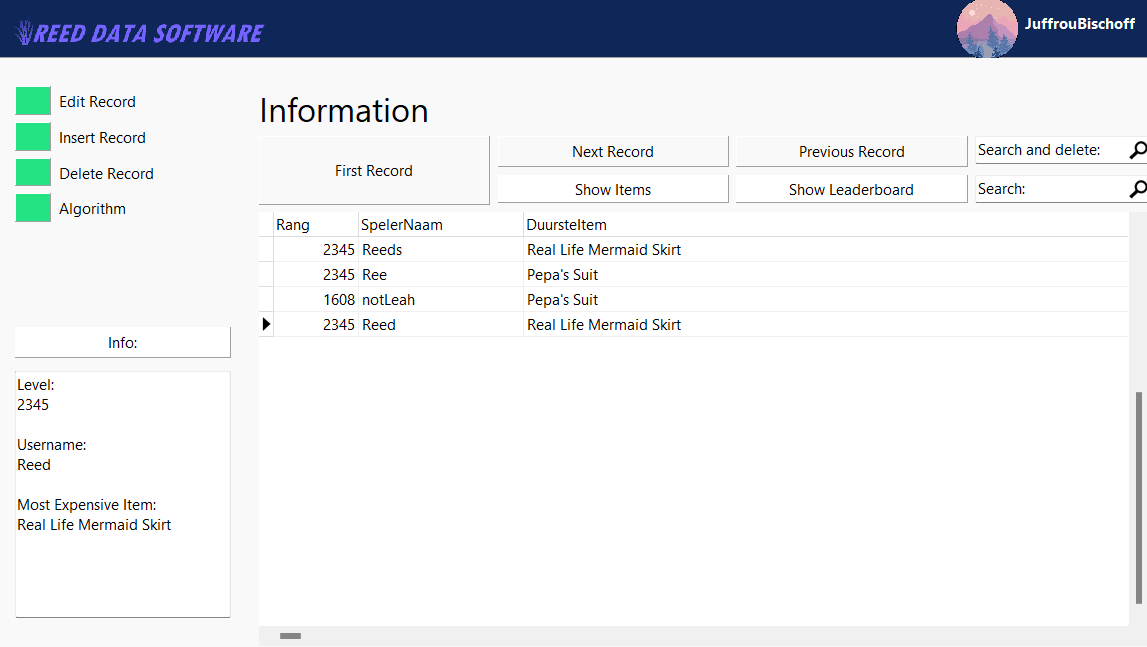
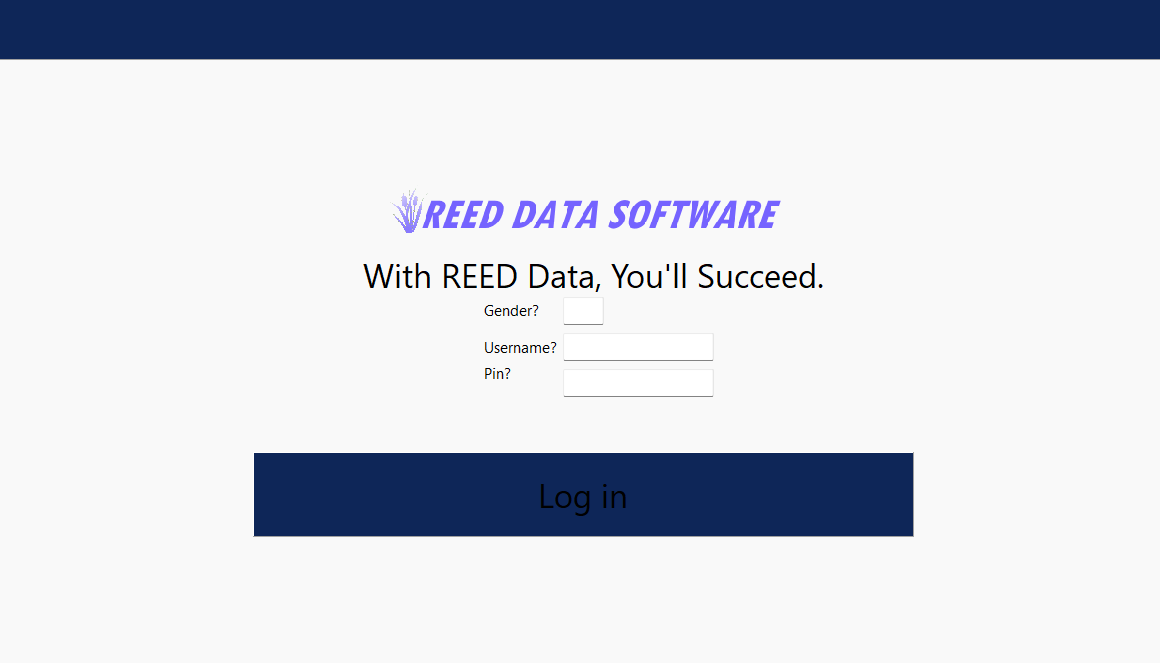
## Databasis gevorderde ontwerp

|  |  |
| --- | --- |
| Relationship van tabelle | |
| Veld in tabel: Item | Veld in tabel: DuursteItem |
| tbl  tbl  **One-to-many relationship.**  Unieke Items in Items tabel. (Primêre sleutel hier)  Unieke SpelerNaam in Ranglys tabel. (Primêre sleutel hier) | |



# Taak 4 – Navigasie

***Form 1: tbsSignIn***

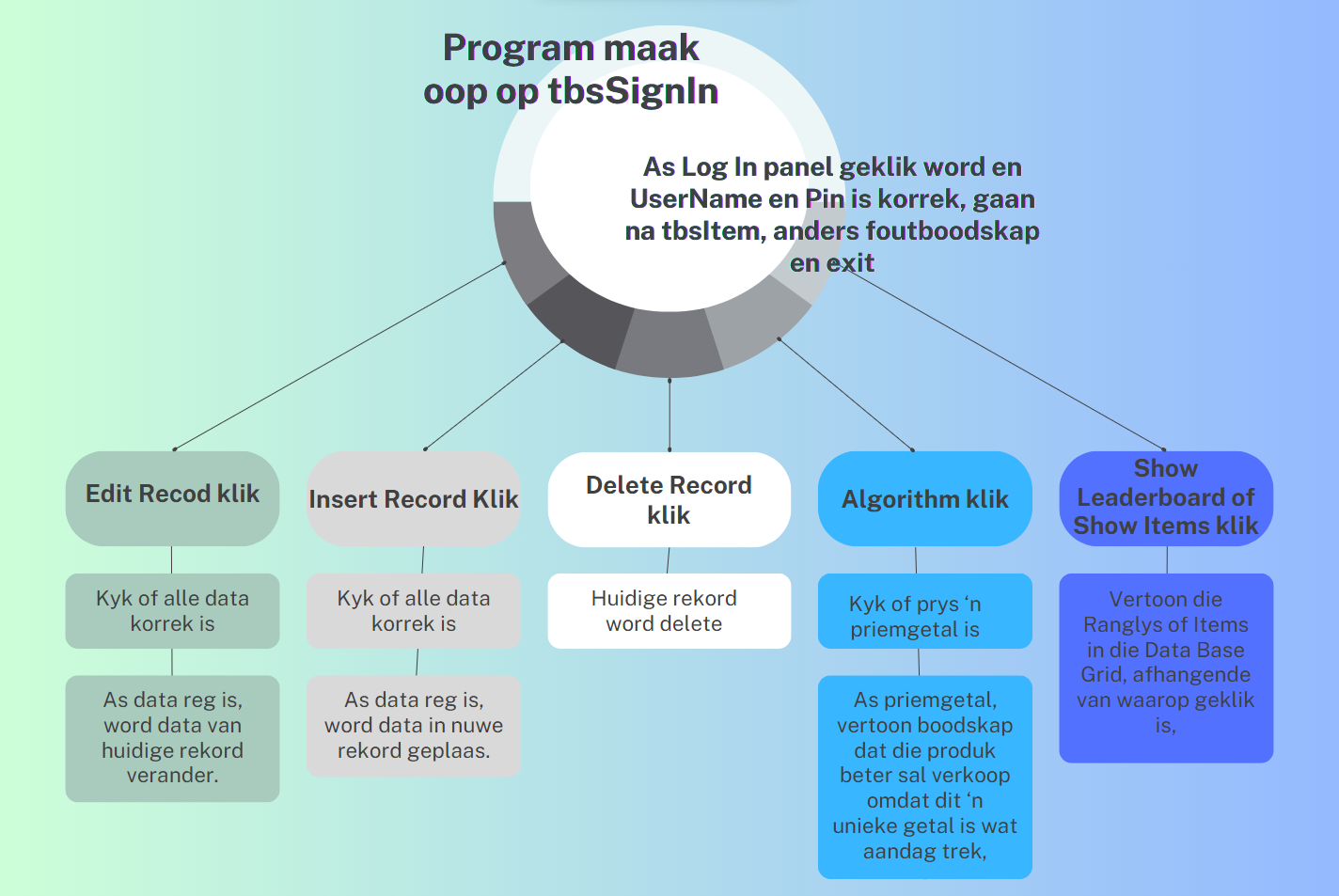
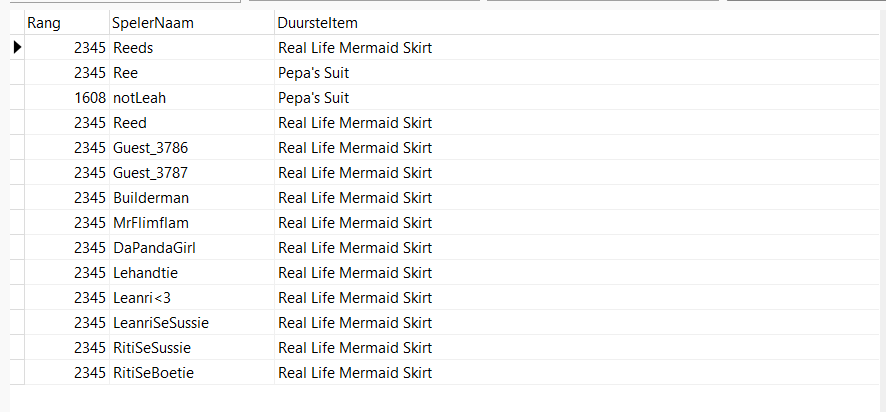
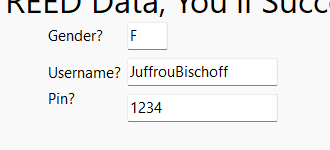


***Form 1: tbsItems***

Die gebruiker klik op die “Log in” panel om te sien of die Username en Pin bymekaar pas, indien iets skeef loop, sal daar ‘n hulpvolle foutboodskap vertoon.

As die “Username” teks en die “pin” –teks dieselfde plekhouer getal het uit die arrays, (arrSpelerNaam en arrPins), sal die gebruiker na die volgende tabsheet toe gevat word. Wanneer die “Log in” getik word.

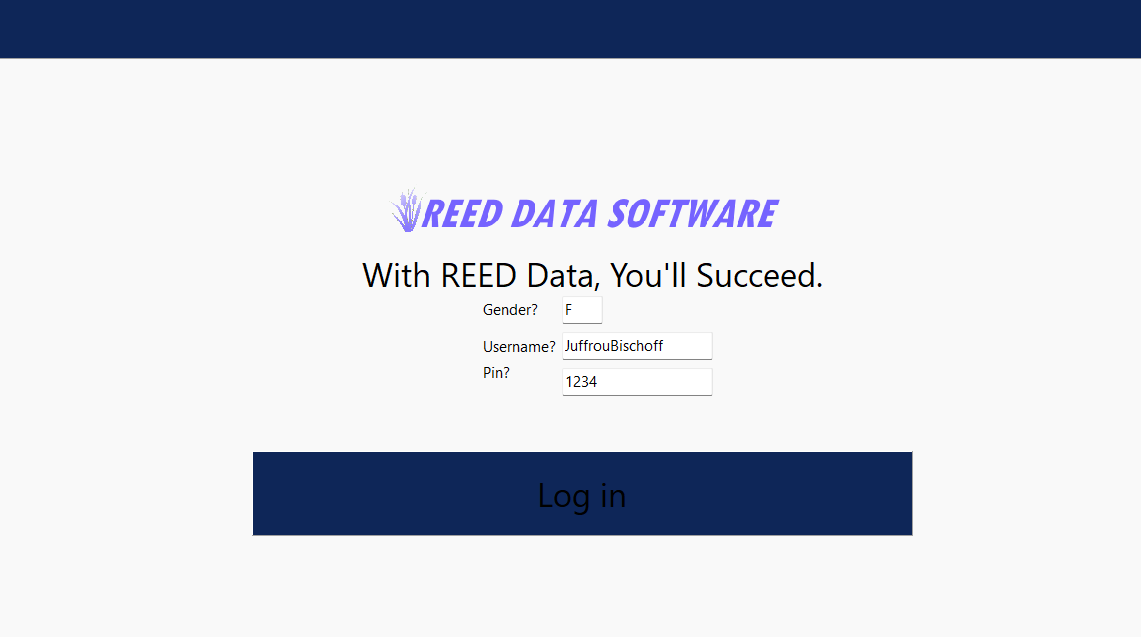
As alles korrek is, by mekaar pas, dan word gebruiker vanaf tbsInteken na tbsAfvoer gevat



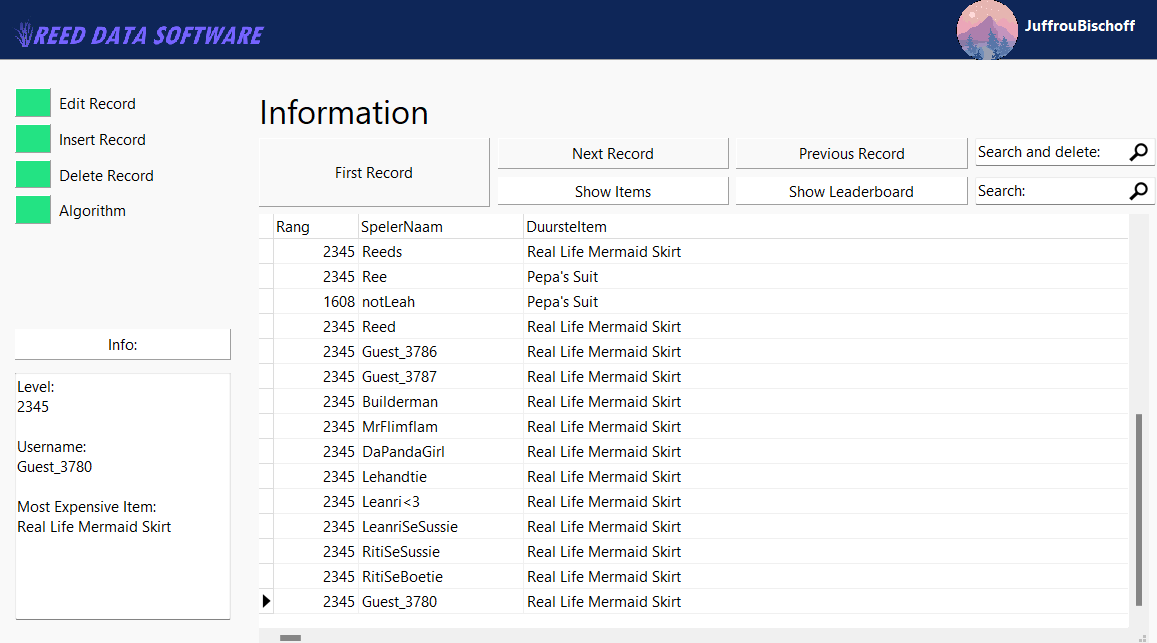
As daar op enige van die knoppies geklik word, word daar algoritmes uitgevoer om die take te voltooi en hulp boodskappe verskyn indien iets skeef loop en die program nie ‘n taak kan uitvoer nie.

# Taak 5 – IPO: Sagteware Ontwerp

|  |  |  |  |
| --- | --- | --- | --- |
| **Form 1 – Tabsheet 1: tbsSignIn** | | | |
| **Invoer** | **Validasie** | **Verwerking //8 Beter voorbeelde op die volgende bladsy.** | **Afvoer** |
| **Data Tipe ontvang:**  Geslag word ingetik as teks (string)  Gebruikersnaam word ingetik as teks (string)  Geheime kode word ingetik as teks (string)  …deur gebruiker | Valideer of geslag **char-waarde** is en of dit = ‘M’ of = ‘F’ is.  **Foutboodskap:** “Gender has to be in M/F format.”  Valideer of Gebruikernaam ‘n teks d.i. **string-waarde** is (nie leeg nie)  **Foutboodskap:** “Username may not be empty.”  Valideer dat die Geheime kode ‘n string-waarde is en net uit 4 karakters bestaan.  **Foutboodskap:** “Pin may only consist of four numbers.”  (Authentication – Kyk of die gebruikersnaam en geheime kode bymekaar pas end at die gebruiker dus is wie hulle eis hulle is)  **Foutboodskap:** “The pin does not match the account.” | 1. Lees die usernames en pins vanuit ‘n teksleer in ‘n array in, in die spesifieke bolgorde sodat die eerste pin en eerste username aan ‘n spesifieke gebruiker behoort. 2. Kyk of die username en pin ooreenstem met mekaar. 3. Indien dit ooreenstem:   Vertoon die volgende tabsheet (tbsItems)  Laai die profile picture op tbsItems  Verander die label op tbsItems wat die gebruikersnaam moet vertoon, sodat dit die korrekte naam vertoon. | **Afvoer data:**  Foutboodskap in lblFaulty word vertoon  Volgende tabsheet word vertoon  Profiel foto word vertoon in die TImage component (imgPfp)  Gebruikersnaam word vertoon in lblWelkom |
| **Bron van data:**  Gebruiker toevoer (Sleutelbord) – Gebruikersnaam en geheime kodes  Teksleer – Gebruikersnaam en geheime kodes |
| **Formaat van Invoer:**  Geslag word gestoor as geslag (M/F)  Gebruikersnaam en geheime kode word gestoor as teks (string) | **Afvoerformaat:**  Teks in label (String)  (foutboodskappe) - String  Visuele afvoer – tbsItems wat verskyn |
| **Komponent gebruik:**  Edit boks om data te ontvang  Panel wat as ‘n knoppie gebruik word | **Komponent gebruik:**  Labels – string (lblFaulty, lblWelkom)  Tabsheet  TImage component (imgPfp) |



|  |  |  |  |
| --- | --- | --- | --- |
| **Form 1 – Tabsheet 2: tbsItems** | | | |
| **TInvoer** | **Validasie** | **Verwerking** | **Afvoer** |
| **Data ontvang:**  Rang van speler  Spelernaam  Speler se duurste item  ***OF***  Item naam  Prys van die item  Seisoen van die item | ***By tblRang:***  Kyk of die rang van die speler uit ‘n **heelgetal (integer**) bestaan.  **Foutboodskap:** “Level has to be an integer value.” En kyk dat die rang net tussen 2 en 2000 is omdat mense nie ‘n hoër rang of laer rang kan hê nie.  Kyk of die spelernaamnaam ‘n string waarde is en nie leeg is of ‘n duplikaat waarde is nie.  **Foutboodskap:** “Username has to be text.” Omdat PlayerName die primary key is…  **Foutboodskap:** “Username cannot be a duplicate value”  Kyk of die duurste item waarde ‘n string waarde is en nie leeg is nie.  **Foutboodskap:** “Most Expensive Item has to be text.”  ***By tblItem:***  Kyk of die Item ‘n string waarde is en nie leeg is nie.  **Foutboodskap:** “Item name has to be a string value” Omdat ItemName die primary key is…  **Foutboodskap:** “Item name cannot be a duplicate value”  Kyk of die prys ‘n **real/float data tipe** is en nie leeg is nie.  **Foutboodskap:** “Price has to be a number value.”  Kyk of ItemSeisoen data ‘n **datum (Tdate)/date tipe** is en ‘n yyyy/mm/dd formaat is.  **Foutboodskap:** “Date has to be in yyyy/mm/dd format.” | 1. **Vertoon alle Spelers.**   dbgInfo.DataSource := dmPAT.dsrRanglys;  bItems := false;  redInfo.Lines.Clear;  with dmPAT do  begin  if bItems = true then  begin  sprys := currtostr(tblItems['Prys']);  rprys := strtofloat(sprys);  sprys := floattostrf(rprys, ffcurrency, 6, 2);  redInfo.Lines.Add('Item: ' + #13 + tblItems['Item'] + #13);  redInfo.Lines.Add('Price: ' + #13 + sprys + #13);  redInfo.Lines.Add('Season: ' + #13 + datetostr(tblItems['ItemSeisoen']));  end  else  begin  redInfo.Lines.Add('Level: ' + #13 + inttostr(tblRanglys['Rang']) + #13);  redInfo.Lines.Add('Username: ' + #13 + (tblRanglys['SpelerNaam']) + #13);  redInfo.Lines.Add('Most Expensive Item: ' + #13 + tblRanglys  ['DuursteItem']);  end;  end;  end;   1. **Vertoon alle Items.**   dbgInfo.DataSource := dmPAT.dsrItems;  bItems := true;  redInfo.Lines.Clear;  with dmPAT do  begin  if bItems = true then  begin  sprys := currtostr(tblItems['Prys']);  rprys := strtofloat(sprys);  sprys := floattostrf(rprys, ffcurrency, 6, 2);  redInfo.Lines.Add('Item: ' + #13 + tblItems['Item'] + #13);  redInfo.Lines.Add('Price: ' + #13 + sprys + #13);  redInfo.Lines.Add('Season: ' + #13 + datetostr(tblItems['ItemSeisoen']));  end  else  begin  redInfo.Lines.Add('Level: ' + #13 + inttostr(tblRanglys['Rang']) + #13);  redInfo.Lines.Add('Username: ' + #13 + (tblRanglys['SpelerNaam']) + #13);  redInfo.Lines.Add('Most Expensive Item: ' + #13 + tblRanglys  ['DuursteItem']);  end;  end;   1. **Voeg rekord by.**   tblItems.insert;  tblItems['Item'] := redInfo.Lines[1];  tblItems['Prys'] := strtofloat(snewprice);  tblItems['ItemSeisoen'] := strtodate(redInfo.Lines[7]);  dmPAT.tblItems.Post;   1. **Verwyder rekord.**   with dmPAT do  begin  if bItems = true then  begin  tblItems.Delete;  // Add to array again...  dbgInfo.DataSource := nil;  dbgInfo.DataSource := dmPAT.dsrItems;  i2 := 0;  tblItems.First;  while not tblItems.Eof and (i2 <= 300) do  begin  inc(i2);  arrItems[i2] := tblItems['Item'];  tblItems.Next;  end;  end;  end;   1. **Verander rekord.**   tblitems.Edit  tblItems['Item'] := redInfo.Lines[1];  tblItems['Prys'] := strtofloat(snewprice);  tblItems['ItemSeisoen'] := strtodate(redInfo.Lines[7]);  tblItems.post   1. **Kyk of die prys ‘n priemgetal is.**   **Algoritme:**  Result := true;  iCount := 0;  if (iNum <= 1) then  begin  Result := false;  end  else  begin  for i := 1 to iNum do  begin  if (iNum mod i = 0) then  begin  inc(iCount);  end;  end;  end;  if (iCount = 2) then  begin  Result := true  end  else  begin  Result := false;  end;   1. **Soek of ‘n rekord in die databasis is.**   **if bItems = true then**  **begin**  **tblItems.First;**  **Algoritme:**  while not tblItems.Eof do  begin // En as tblRanglys nou vertoon word, word die algoritme aangepas om ‘n spelernaam te vind!  if **tblItems['Item'] = sbxSoek.Text** then  begin  **showmessage('Record found.');**  **tblItems.Delete;**  **exit↘️;**  end  else  begin  **tblItems.Next;**  end;  end;   1. **Vertoon ‘n rekord se informasie wanneer daar op ‘n cell geklik word.**   redInfo.Lines.Clear;  with dmPAT do  begin  if bItems = true then  begin  sprys := currtostr(tblItems['Prys']);  rprys := strtofloat(sprys);  sprys := floattostrf(rprys, ffcurrency, 6, 2);  redInfo.Lines.Add('Item: ' + #13 + tblItems['Item'] + #13);  redInfo.Lines.Add('Price: ' + #13 + sprys + #13);  redInfo.Lines.Add('Season: ' + #13 + datetostr(tblItems['ItemSeisoen']));  end  else  begin  redInfo.Lines.Add('Level: ' + #13 + inttostr(tblRanglys['Rang']) + #13);  redInfo.Lines.Add('Username: ' + #13 + (tblRanglys['SpelerNaam']) + #13);  redInfo.Lines.Add('Most Expensive Item: ' + #13 + tblRanglys  ['DuursteItem']);  end;  end; | **Afvoer data:**  Boodskap verskyn wat vir die gebruiker sê dat die item se prys ‘n priemgetalwaarde is of nie.  tblItems of tblRanglys verskyn in dbGrid.  Databasis word verander. -> Gekose aksie vind plaas.  Informasie van die rekord word in rich edit vertoon. |
| **Bron van data:**  Gebruiker toevoer – Sleutelbord en muis, nuwe data in te tik of verander  Databasis om rekords te verskaf (en te behou) |
| **Formaat van Invoer:**  ItemSeisoen word gestoor as datum, prys as geldeenheid, rang teks of integer, naam van speler en items as teks | **Afvoerformaat:**  Geldeenheid – Currency  (ItemPrys)  Datum – Tdate (ItemSeisoen)  Teks – string  (Item naam, speler naam, Duurste Item)  Integer (Level) |
| **Komponent gebruik:**  Rich edit om data te ontvang om te verander, in te voeg of te verwyder  Search Box  Panel as knoppies om te weet watter verwerking moet plaasvind. | **Komponent gebruik:**  Rich Edit – String (Informasie van Item of speler)  dbGrid – Verskeie data tipes  (Wys tblItems en tblRanglys)  Message – String  (Sê of die prys se heelgetal waarde ‘n priemgetal is of nie)  Textfile – String |



# Taak 6 – HCI Principles: GUI Ontwerp

// Ek het 3 tabsheets, al drie op frmPAT.

|  |  |
| --- | --- |
| **Form 1 – Tabsheet 1:** tbsSignIn | |
| HELP | Doel van program en gebruiker:  Die gebruiker wil hierdie tabsheet gebruik om in te teken om daarna toegang te verkry om sy data, in ‘n databasis gestoor, te verander. Die tabsheet het ‘n eenvoudige en modern uitleg om daardie spesifieke taak te verrig. Daar is ‘n konstante ontwerp patroon deur die hele program met die donker blou forms en logo. Die font van die teks bly ook konstant en is maklik om te lees. Dit neem nie te veel tyd van die gebruiker op nie. |
| Vriendelike dialoog:  Indien iemand op die ‘HELP’ paneel klik, sal ‘n boodskap verskyn wat hulp aan die gebruiker bied. Dit sal verduidelik dat die gebruikers hulle name en kodes moet insleutel om in te teken en dat hul geslag in spesifieke formaat gestoor moet word. Die geslag word net gevra om toekomstige advertensies in te voer op grond van gebruikers se behoeftes. |
| Standaard GUI ontwerpbeginsels:  Ek het die komponente gekies op grond van gemak vir die gebruiker en spoed wat hulle kan inteken. Die editbokse is gekies omdat dit maklik in strings ingelees kan word en getoets kan word of die gebruikersnaam en kode bymekaar pas. Dit is maklik en vinnig vir die gebruiker om alle data in die editbokse te tik en so in te teken. Die inteken knoppie en Hulp-knoppie is panels omdat die kleur van panels verander kan word vir ‘n beter gebruikerservaring. | |
| Gebruiksgemak, logiese vloei:  Die cursor sal van die geslag boks outomaties oorbeweeg na die username edit boks indien die veld se inligting korrek is en so ook na die pin edit boks toe. Die gerbuiker kan gemaklik na die volgende skerm toe gaan deur die log in te klik. Indien die gebruiker op die hulp-boks kliek, word ‘n hulp boodskap gewys en daar is ‘n kruisie wat gedruk kan word om hom te laat af vlieg. Daar is baie min dele waar die gebruiker toevoer hoef te gee, wat my program gerieflik maak. | |
| Duidelik gemerkte navigasie:  Alle navigasie komponente is duidelik sigbaar en het net een doel wat duidelik bekend is aan die gebruiker. Daar is ‘n hulp knoppie (panel) wat verduidelik op watter inligting ingevoer moet word om die gewenste resultate te kry en verduidelik op watter ander knoppies (panels) geklik moet word. | |
| Nuttige foutboodskappe:  Indien toevoer van ‘n verkeerde formaat soos letters waar nommer moet wees of daar op ‘n plek geen toevoer gegee word nie, sal ‘n foutboodskap verskyn wat presies verduidelik wat die gebruiker ingetik het en wat die gerbuiker eintlik moet intik. | |

|  |  |  |
| --- | --- | --- |
| ***Taak*** | ***Komponent*** | ***Gebeurtenis*** |
| *Vra ‘n vraag aan die gebruiker.* | *lblGeslag, lblGebruikersnaam, lblPin* | *Geen gebeurtenis* |
| *Vertoon foutboodskap.* | *lblFaulty* | *(Wanneer pnlLogIn gekliek is word ‘n unieke foutboodskap hier vertoon wat spesifiek is tot die gebruiker.)* |
| *Maak die program deftiger.* | *imgLogo, lblMyWoorde, pnlHeader* | *Geen gebeurtenis* |
| *Vertoon hulp boodskap wat nuttig is vir die gebruiker en verduidelik wat die gebruiker moet doen in die vorm van ‘n opwipboodskap.* | *pnlHelp* | *OnClick event* |
| *Dien as ‘n knoppie om te toets of die gebruikersnaam en kode ooreenstem.* | *pnlLogIn* | *OnClick event* |

|  |  |
| --- | --- |
| **Form 1 – Tabsheet 2:** tbsItems | |
| HELP  BACK  More Products | Doel van program en gebruiker:  Die gebruiker wil hierdie tabsheet gebruik om sy data, in ‘n databasis gestoor, te verander of te verwyder. Die tabsheet het ‘n eenvoudige en modern uitleg om daardie spesifieke taak te verrig. Daar is ‘n konstante ontwerp patroon deur die hele program met die donker blou forms en logo. Die font van die teks bly ook konstant en is maklik om te lees. Dit neem nie te veel tyd van die gebruiker op nie. |
| Vriendelike dialoog:  Indien iemand op die ‘HELP’ paneel klik, sal ‘n boodskap verskyn wat hulp aan die gebruiker bied. Dit sal verduidelik watter toevoer verwag word en in watter formaat die datum moet wees. Dit help ook die gebruiker sien wat hy alles in staat is om te kan doen deur my program te gebruik hulpvolle foutboodskappe word ook deurgegee. |
| Standaard GUI ontwerpbeginsels:  Ek het die komponente gekies op grond van gemak vir die gebruiker en spoed wat hulle kan inteken. Die richedit is gekies omdat dit maklik in strings ingelees kan word en die data gevalideer en geformatteer word. Dit lyk ook meer professioneel as aparte komponente nie geskik vir verkeie datatipes nie. Dit is maklik en vinnig vir die gebruiker om alle data in dieselfde komponent te tik en so gewenste resultate te kry. Die knoppies is almal panels omdat die kleur van panels verander kan word vir ‘n beter gebruikerservaring. | |
| Gebruiksgemak, logiese vloei:  Die gerbuiker kan gemaklik na die vorige skerm toe gaan deur “BACK” knoppie in te klik en na die volgende skerm deur “More products” te klik, end an weer terug gaan ook! Indien die gebruiker op die hulp-boks kliek, word ‘n hulp boodskap gewys en daar is ‘n kruisie wat gedruk kan word om hom te laat af vlieg. Daar is baie min dele waar die gebruiker toevoer hoef te gee, wat my program gerieflik maak. Die vloei is chronologies, logies en word goed beskryf. Mense kan van enige tabsheet na enige ander tabsheet toe gaan met twee of minder kliks. Die cursor word outomaties geskyf na die volgende component of rekord wat met die data te doen het. | |
| Duidelik gemerkte navigasie:  Die exit-blokke is duidelik gemerk. Alle navigasie komponente is duidelik sigbaar en het net een doel wat duidelik bekend is aan die gebruiker. Daar is ‘n hulp knoppie (panel) wat verduidelik watter inligting ingevoer moet word om die gewenste resultate te kry en verduidelik op watter ander knoppies (panels) geklik moet word. | |
| Nuttige foutboodskappe:  Indien toevoer van ‘n verkeerde formaat soos letters waar nommer moet wees, datum in die verkeerde formaat of daar op ‘n plek geen toevoer gegee word nie, sal ‘n foutboodskap verskyn wat presies verduidelik wat die gebruiker ingetik het en wat die gerbuiker eintlik moet intik. | |

|  |  |  |
| --- | --- | --- |
| ***Taak*** | ***Komponent*** | ***Gebeurtenis*** |
| *Maak die program deftiger.* | *pnlHeader, imgLogo, lblInformation, pnlHeadeingRed, imgPfp, lblName* | *Geen gebeurtenis* |
| *Vertoon hulp boodskap wat nuttig is vir die gebruiker en verduidelik wat die gebruiker moet doen in die vorm van ‘n opwipboodskap.* | *pnlHelp* | *OnClick event* |
| *Takes to tbsSignIn when pnlBack is clicked and tbsMessage when pnlMoreProducts is clicked.* | *pnlMoreProducts, pnlBack* | *OnClick* |
| *Navigate to first, next and previous record, depending on the name of the panel.* | *pnlFirst, pnlNext, pnlPrevious* | *OnClick* |
| *Soek ‘n speler/item en vertoon ‘n boodskap as die speler/item in die databsis is. Soek ‘n speler/item en delete die rekord as die speler/item in die databsis is.* | *sbxDelete, sbxSearch* | *OnChange* |
| *Algoritmes word uitgevoer om ‘n rekord te verander, by te sit of te delete vanaf die databasis af. (Afhangende van die naam van die paneel waarop geklik word)* | *pnlEdit, pnlInsert, pnlDelete, pnlAlgorithm* | *OnClick* |
| *Die informasie van die rekord word vertoon (die rekord waarop gebruiker gekliek het)* | *redDisplay* | *(Works with OnCellClick from dbgItemsnPlayers)* |
| *Vertoon al die Items of spelers afhangende van waarop gekliek is (pnlPlayers of pnlItems)* | *dbgItemsnPlayers* | *OnCellClick* |
| *Vertoon alle spelers of items in die databasis afsonderlik* | *pnlPlayers, pnlItems* | *OnClick* |

|  |  |
| --- | --- |
| **HCI-Beginsel** | **Hoe kom my skerm dit na? (‘n Voorbeeld word verskaf, maar my program het vele meer boodskappe, almal soortgelyk.)** |
| 1. Geoeie gebruik van spasie op skerm. | Die uitleg is so beplan dat komponente duidelik is en soortgelyke funksies bymekaar gebruik kan word. Die oop spasie is gebruik om die funksies van my program uit te hef. |
| 1. Goeie gebruik van kleur op skerm. | Die kleure is hoogs kontrasterend sodat swak-siende gebruikers my program met gemak kan gebruik. |
| 1. Nuttige foutboodskappe. | Al my foutboodskappe sê duidelik aan die gebruiker wat verkeerd gedoen is en wat hy moet doen. |
| 1. Bruikbaarheid en gebruikersbenodighede voldoende. | Die tabsheet maak die doel van die gebruiker maklik bereikbaar. Die uitleg is nie moeilik om te navigeer nie en die hulp-fasiliteite gee presiese stappe oor wat gedoen moet word om die gewensde resultate te kry. |
| 1. Terugvoer is netjies, duidelik en goed voorgestel. | My boodskappe gee terugvoer sodat die gebruiker weet wat plaasvind in die program. |

|  |  |
| --- | --- |
| **Form 1 – Tabsheet 3:** tbsMessage | |
| HELP  BACK | Doel van program en gebruiker:  Die gebruiker wil hierdie tabsheet gebruik om nog produkte te koop wat dieselfde tipe GUI-uitleg as my program het. Die tabsheet het ‘n eenvoudige en modern uitleg om daardie spesifieke taak te verrig. Daar is ‘n konstante ontwerp patroon deur die hele program met die donker blou forms en logo. Die font van die teks bly ook konstant en is maklik om te lees. Dit neem nie te veel tyd van die gebruiker op nie. |
| Vriendelike dialoog:  Indien iemand op die ‘HELP’ paneel klik, sal ‘n vriendelike boodskap verskyn wat hulp aan die gebruiker bied. Dit sal verduidelik dat die gebruikers op die foto moet klik om na die webstuiste op die internet te gaan, anders moet die gebruiker terugkeer na die hoof koppelvlak toe deur “BACK” te druk. |
| Standaard GUI ontwerpbeginsels:  Ek het die komponente gekies op grond van gemak van die gebruiker. Die Image component is gekies om ‘n foto te wys en die panele om die program meer aantreklik te maak. Dit is maklik en vinnig vir die gebruiker om meer produkte aanlyn te koop deur die Image op te kliek. (Die image neem deur gebruik vam ‘n link die gerbuiker na ‘n webwerf toe) Die terug knoppie en Hulp-knoppie is panels omdat die kleur van panels verander kan word vir ‘n beter gebruikerservaring. En ‘n label word gerbuik om teks op ‘n modern manier voor te stel sonder ‘n boks of vorm van ‘n agtergrond. Die tabsheet gee terugvoer aan die persoon en die uitleg bly konstant en eenvoudig. | |
| Gebruiksgemak, logiese vloei:  Die gerbuiker kan gemaklik na die vorige skerm toe gaan deur “BACK” knoppie in te klik. Indien die gebruiker op die hulp-boks kliek, word ‘n hulp boodskap gewys en daar is ‘n kruisie wat gedruk kan word om hom te laat af vlieg. Daar is baie min dele waar die gebruiker toevoer hoef te gee, wat my program gerieflik maak. Die vloei is logies. Mense kan van enige tabsheet na enige ander tabsheet toe gaan met twee of minder kliks. | |
| Duidelik gemerkte navigasie:  Alle navigasie komponente is duidelik sigbaar en het net een doel wat duidelik bekend is aan die gebruiker. Daar is ‘n hulp knoppie (panel) wat verduidelik op watter inligting ingevoer moet word om die gewenste resultate te kry en verduidelik op watter ander knoppies (panels) geklik moet word. | |
| Nuttige foutboodskappe:  Indien gebruiker nie gekonnekteer aan die internet is nie of die webstuiste laai nie, word ‘n nuttige foutboodskap vertoon wat verduidelik wat die gebruiker moet doen en dat die webtuiste nie kon laai nie in die vorm van ‘n opwip boodskap. | |

|  |  |  |
| --- | --- | --- |
| ***Taak*** | ***Komponent*** | ***Gebeurtenis*** |
| *Neem die gebruiker na die webstuiste.*  *Wys die gebruiker die besigheid se naam, sodat die gebruiker produkte van die besigheid kan kry.* | *imgLink* | *OnClick* |
| *Vertoon teks wat verduidelik wat op die webstuiste is.* | *lblLinkInfo* | *Geen gebeurtenis* |
| *Maak die program deftiger.* | *pnlHearder* | *Geen gebeurtenis* |
| *Neem die gebruiker na die vorige skerm wat gewys het (tbsItems)* | *pnlBack* | *OnClick event* |
| *Vertoon hulp boodskap wat nuttig is vir die gebruiker en verduidelik wat die gebruiker moet doen in die vorm van ‘n opwipboodskap.* | *pnlHelp* | *OnClick event* |

|  |  |
| --- | --- |
| **HCI-Beginsel** | **Hoe kom my skerm dit na? (‘n Voorbeeld word verskaf, maar my program het vele meer boodskappe, almal soortgelyk.)** |
| 1. Geoeie gebruik van spasie op skerm. | Die uitleg is so beplan dat komponente duidelik is en soortgelyke funksies bymekaar gebruik kan word. Die oop spasie is gebruik om die funksies van my program uit te hef. |
| 1. Goeie gebruik van kleur op skerm. | Die kleure is hoogs kontrasterend sodat swak-siende gebruikers my program met gemak kan gebruik. |
| 1. Nuttige foutboodskappe. | Al my foutboodskappe sê duidelik aan die gebruiker wat verkeerd gedoen is en wat hy moet doen. |
| 1. Bruikbaarheid en gebruikersbenodighede voldoende. | Die tabsheet maak die doel van die gebruiker maklik bereikbaar. Die uitleg is nie moeilik om te navigeer nie en die hulp-fasiliteite gee presiese stappe oor wat gedoen moet word om die gewensde resultate te kry. |
| 1. Terugvoer is netjies, duidelik en goed voorgestel. | My boodskappe gee terugvoer sodat die gebruiker weet wat plaasvind in die program. |