NEW MEDIA &
COMMUNICATION
TECHNOLOGY

Business applications
Project cashless payment

Algemeen

Elke week zal je een deel van je project moeten afwerken. Het deel dat moet afgewerkt zijn zal telkens beschreven staan bij de Todo. Er zal niet elke week voor iedereen een evaluatie van het project zijn, het is dus je eigen verantwoordlijkheid om te controleren of je nog op schema zit. Tijdens het labo zullen wel elke week ongeveer 5 studenten uitgekozen worden om kort de stand van zaken rond hun project toe te lichten. Zorg er dus voor dat je altijd een recente versie van het project hebt staan tijdens het labo.

Evaluatie

Dit project telt voor 40% mee in het eindcijfer van de module**. Belangrijk om te vermelden is dat deze 40% niet hernomen kan worden in de tweede zittijd.** Je neemt dus het punt van de eerste zittijd mee.

Indienen

Er zijn verschillende tussentijdse deadlines waar op je het project moet indienen. Kijk aandachtig bij de Todo sectie, daar staat telkens aangegeven of je het project al dan niet op de dropbox van Leho moet indienen.

Projectomschrijving

Algemeen

Als project voor deze module maak je een applicatie dat het cashless payment systeem simuleert dat onder andere bij voetbalploegen als Club Brugge en Borussia Dortmund wordt gebruikt. Een voorbeeld van hoe dergelijk systeem werkt kan je bekijken op http://clubbrugge.be/nl/nieuws/18498/cashless-payment-op-club-brugge-instructie-video.

Stakeholders

Er zijn vier verschillende stakeholders in dit project die elk hun rol spelen.

IT bedrijf

Het IT bedrijf zal de software ontwikkelen en voor zichzelf een applicatie voorzien die het mogelijk maakt om nieuwe gebruikers voor het systeem aan te maken, kassa's toe te kennen en om via een logboek fouten in het systeem op te sporen. Daarnaast levert het IT bedrijf ook de hardware voor de kassa systemen.

Vereniging: management

Een vereniging kan een voetbalploeg zijn, maar ook een cultureel centrum of gelijk welke organisatie dat een kassasysteem nodig heeft. De vereniging moet de mogelijkheid hebben om een prijskaart aan te maken van hun producten, een overzicht van hun klanten te hebben en statistieken in verband met de verkoop te bekijken. Het moet ook mogelijk zijn om een lijst met medewerkers bij te houden zodat er geweten is wie op welk moment een kassa heeft bemand.

Vereniging: medewerkers

De medewerkers van de vereniging zijn diegene die op evenementen gebruik zullen maken van het kassasysteem. Zij moeten de mogelijkheid hebben om de bestelling van een klant op een vlotte manier af te handelen.

Klant

De klant is een bezoeker van een evenement die via het cashless payment systeem een aankoop kan doen. Daarnaast moet het voor een klant ook mogelijk zijn om op een vlotte manier zijn kaart op te laden.

Requirements

IT bedrijf (A)

- Beheer verenigingen (A1)
  - Kan een nieuwe vereniging registreren met een login en een wachtwoord (A11)
  - Kan een database genereren voor een nieuwe vereniging (A12)
  - Kan basisinformatie over een vereniging invoeren (A13)
  - Kan basisinformatie over een vereniging bewerken (A14)
  - Kan basisinformatie over een vereniging weergeven (A15)
  - Kan een lijst van verenigingen die het systeem gebruiken weergeven (A16)
- Beheer kassa's (A2)
  - Kan een nieuwe kassa toevoegen aan zijn assortiment (A21)
  - Kan een kassa toekennen aan een vereniging (A22)
  - Kan een kassa wijziging van vereniging (A23)
  - Kan een overzicht van alle kassa's opvragen per vereniging (A24)
  - Kan een overzicht van alle beschikbare kassa's opvragen (A25)
- Logging (A3)
  - Kan een logboek van foutmeldingen en waarschuwingen van de kassa's aanmaken (A31)
  - Kan een logboek van foutmeldingen en waarschuwingen van de kassa's opvragen (A32)

Vereniging: management (B)

- Beheer account (B1)
  - Kan zich aanmelden op zijn account (B11)
  - Kan zich afmelden van zijn account (B12)
  - Kan zijn paswoord wijzigen (B13)
- Beheer producten (B2)
  - Kan een product toevoegen aan een prijslijst (B21)
  - Kan de prijs van een product wijzigen uit de prijslijst (B22)
  - Kan een product verwijderen uit de prijslijst (B23)
- Beheer medewerkers (B3)
  - Kan een lijst met medewerkers weergeven (B31)
  - Kan een medewerker aan de lijst toevoegen (B32)
  - Kan gegevens van een medewerker wijzigen (B33)
  - Kan een medewerker uit de lijst verwijderen (B34)
- Beheer kassa (B4)
  - Kan een overzicht van al zijn kassa's weergeven (B41)
  - Kan opvragen welke medewerker een kassa bemand heeft (B42)
- Beheer klanten (B5)
  - Kan een overzicht van al zijn klanten weergeven (B51)
  - Kan gegevens (ook het saldo) van een klant wijzigen (B52)
- Beheer statistieken (B6)
  - Kan de totale verkoop opvragen over een bepaalde periode (B61)
  - Kan de verkoop per kassa opvragen over een bepaalde periode (B62)
  - Kan de verkoop per product opvragen over een bepaalde periode (B63)

Vereniging: medewerker (C)

- Kan zichzelf identificeren (C1)
- Kan een klant identificeren (C2)
- Kan het saldo van een kaart weergeven (C3)
- Kan een bestelling ingeven (C4)

Klant (D)

- Kan zichzelf registreren om het systeem te gebruiken (D1)
- Kan zijn kaart opladen voor een bedrag van maximum 100 euro (D2)
- Kan het huidige saldo van zijn kaart controleren (D3)

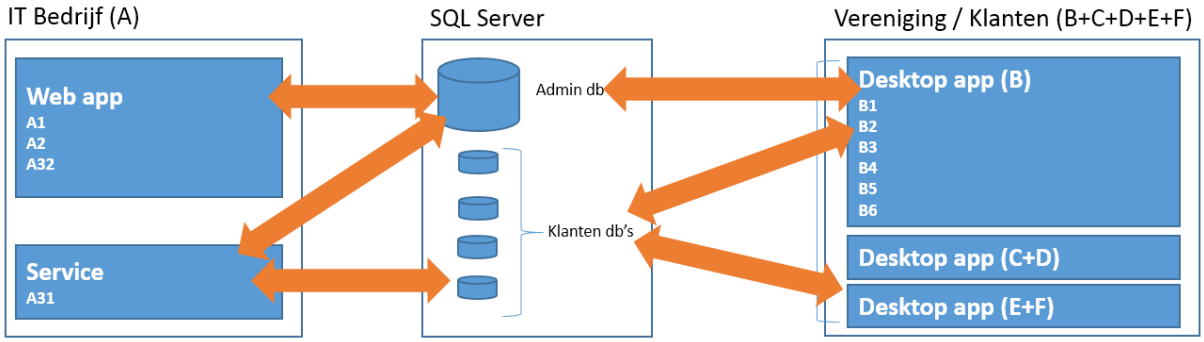
Kassasysteem: medewerkers (E)

- Kan een waarschuwing geven als de prijs van de bestelling groter wordt dan het saldo van de kaart (E1)
- Kan het saldo van de kaart aanpassen (E2)
- Kan de bestelling opslaan voor de statistieken (E3)
- Kan foutmeldingen en waarschuwingen opslaan in het logboek (E4)

Kassasysteem: klant (F)

- Kan controleren of een klant al geregistreerd is (F1)
- Kan biljetten van 5, 10, 20 en 50 euro herkennen (F2)
- Kan foutmeldingen en waarschuwingen opslaan in het logboek (F3)

Architectuur



Het project zal gebruik maken van verschillende databases. Er is een database waar het IT bedrijf al zijn data zoals gegevens van verenigingen en kassa's zal bijhouden. Daarnaast beschikt elke vereniging over zijn eigen database.

Voor de noden van het IT bedrijf wordt er een ASP.NET MVC web applicatie uitgewerkt. Daarnaast zal het IT bedrijf gebruik moeten maken van een component of service die de log van de kassa's uit de database van de klant haalt en importeert in zijn eigen database.

Voor de vereniging en de klanten wordt er gewerkt met desktop applicaties. Deze applicaties zullen communiceren met de database van de vereniging zelf. Het is dus niet de bedoeling dat verenigingen de database van het IT bedrijf constant zullen raadplegen. De enige uitzondering is om aan te melden bij Desktop app B. Hiervoor worden de nodige gegevens uit de admin db gehaald.

De web applicatie wordt uitgewerkt in de module Server Side Application, de desktop applicaties binnen de module Business Applications.

Stap 1: het tekenen van de grafische user interface

Tijdens deze eerste stap worden de grafische user interfaces voor het project ontworpen op papier. Lees eerst aandachtig door de opsomming van de verschillende requirements en schets enkele user interfaces op papier.

Kies daarna de user interface uit die je voorkeur geniet en werk deze in het net uit. Controleer zeker of deze user interface alle gevraagde requirements kan invullen.

Tip

Bij het ontwerpen van de user interface moet je altijd de context in gedachten houden waar binnen de applicatie wordt gebruikt. In dit geval moet het voor de medewerkers tijdens een evenement vlot mogelijk zijn om een bestelling in te geven en ook de klant moet met weinig moeite zijn kaart kunnen opladen. Daarnaast is het ook niet evident om overal een klavier te voorzien met een muis. Een user interface die gebruik maakt van touch moet op zijn minst overwogen worden.

Todo

Maak een user interface op papier voor de gevraagde requirements. Dit doe je zowel voor de web applicatie als voor de desktop applicaties.

Je dient een kopie in van je papieren versie voor de start van het volgende labo (week van 29/9 tot 3/10). Je kan dit doen in het Pand of je geeft af aan je labo docent net voor de start van het labo. Zorg er natuurlijk wel voor dat je zeker nog een kopie of het origineel hebt voor jezelf.

Stap 2: wireframes omzetten naar XAML

Tijdens de theorie en het labo heb je geleerd hoe je in XAML een user interface kan aanmaken en hoe je opmaak kan voorzien door gebruik te maken van verschillende styles. De bedoeling in de tweede stap is om deze kennis te gebruiken om het wireframe om te zetten naar code.

Je zal hiervoor een nieuw project moeten aanmaken in Visual Studio. Volg hiervoor de volgende stappen:

- Maak een blanco solution aan met als naam nmct.ba.cashlessproject
- Voeg een nieuw WPF project toe met als naam nmct.ba.cashlessproject.tempui

Tip

Gebruik zo veel mogelijk een grid als container om je elementen in te plaatsen. Een grid zal er namelijk voor zorgen dat alle onderdelen mooi mee resizen als de afmetingen van het Window worden aangepast.

In de module Server Side Applications heb je ook gezien hoe je Git kan gebruiken bij je projecten. Zorg dan ook dat je een Git repository aanmaakt voor de volledige solution.

Todo

Maak voor elk scherm uit je wireframe een nieuw Window aan in een WPF project. Bij WPF worden normaal gezien user controls in een Window geplaatst om te navigeren. Deze stap zal in het project pas later genomen worden.

Je hoeft je project **niet** in te dienen, maar zorg dat je het in **het labo van 6/10 tot 10/10** zeker bij je hebt om te kunnen voorleggen aan de labo docent.

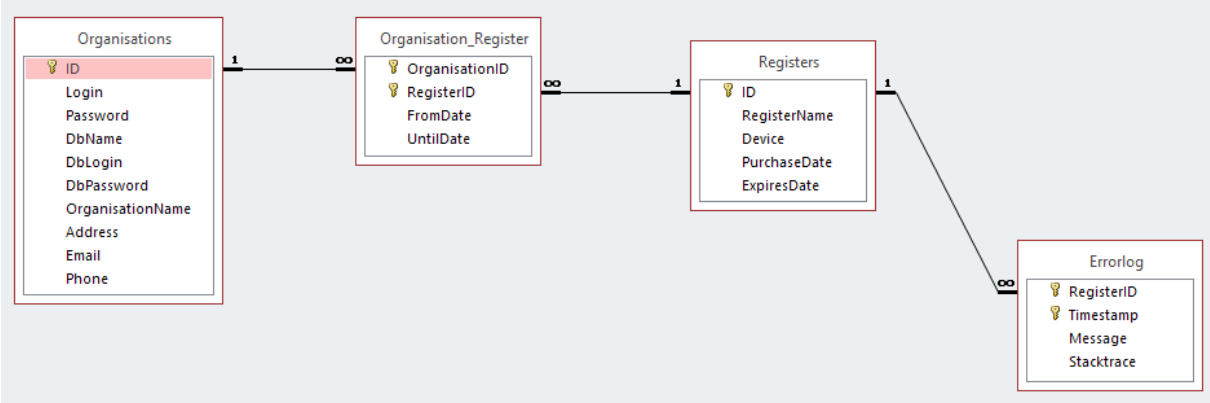
Stap 3: ontwikkelen van het model

Bij deze stap maak je de nodige classes aan voor het model. Dit zijn de classes die de structuur van de data zullen beschrijven en de methods zullen bevatten om deze data op te halen of te bewerken.

Voorlopig beperken we ons tot het aanmaken van de classes en de properties. De methods zullen pas later aan bod komen.

De models zullen aangemaakt worden op basis van de ER-diagramma's. De algemene regel die je kan hanteren is dat elke tabel een class moet worden en elk veld in de tabel een property. Uitzondering zijn tussentabellen. Voor deze tabellen hoeft er geen klasse aangemaakt te worden aangezien ze aangesproken zullen worden door één van de hoofdtabellen.

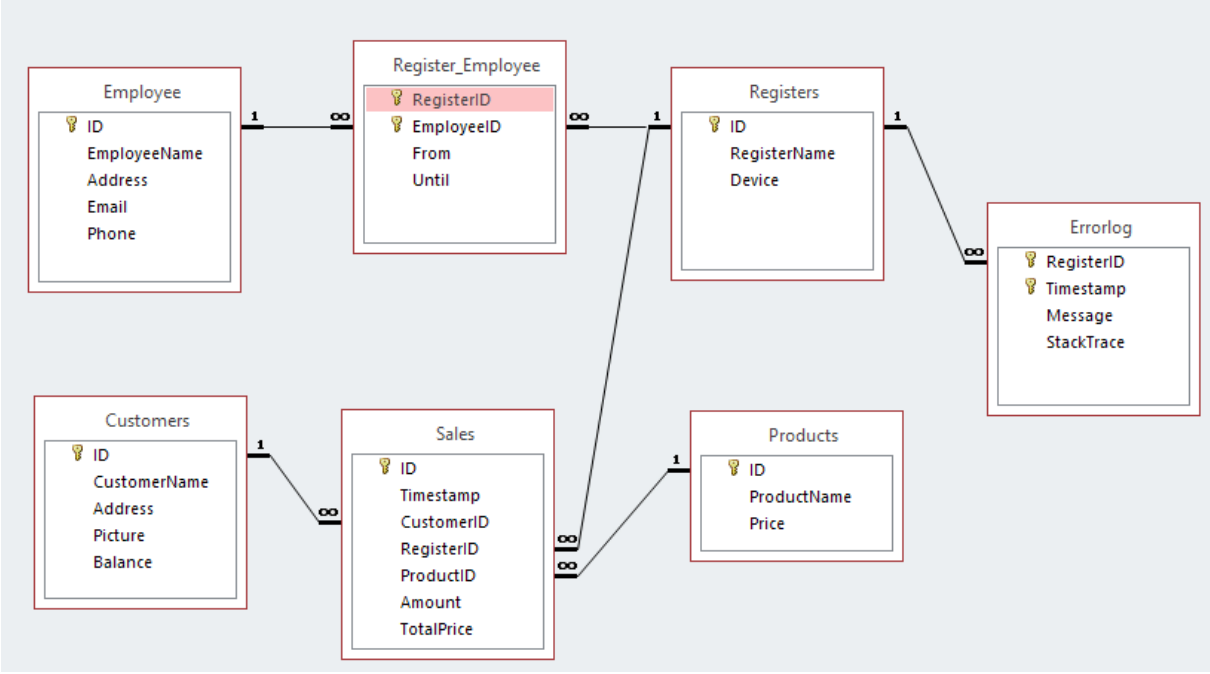
ER diagramma's



Het schema hierboven toont het ER-diagramma voor de centrale database van het IT-bedrijf. De centrale tabellen zijn Organisations en Registers. De n op n relatie tussen deze tabellen vereist de tussentabel Organisation_Register die kan bijhouden welke vereniging welke kassa's in zijn bezit heeft voor een bepaalde periode. Indien de UntilDate niet gekend is wordt automatisch de ExpiresDate uit de tabel Registers overgenomen. Dit is de datum waarop de kassa afgeschreven is. Een periode van vijf jaar na aankoop is hier realistisch.

De tabel Organisations bevat velden om aan te melden bij de desktop applicatie (Login en Password) en ook velden om connectie te maken met de specifieke database van een organisatie (DbName, DbLogin en DbPassword). Deze kunnen echter niet door de klant aangepast worden. De klant kan echter wel zijn eigen paswoord aanpassen.

Tot slot is er nog de tabel ErrorLog. Deze zal alle foutmeldingen bevatten die een kassa bij een vereniging heeft gegenereerd.



Het tweede schema toont de database die voor elke klant wordt aangemaakt. De hoofdtabellen zijn:

- Employee: basisinformatie over een medewerker
- Customer: basisinformatie over de klanten
- Products: de naam en prijs van een product
- Registers: info over de kassa's die toegekend werden aan een klant

De tussentabel Register_Employee biedt de mogelijkheid om te controleren welke medewerker een bepaalde kassa bediend heeft tussen 2 tijdstippen. Bijvoorbeeld op 10/10/2014 van 18u tot 20u.

Centraal staat de tabel Sales die de info van een verkoop zal bevatten. De timestamp bevat het exacte tijdstip van de verkoop (zie tips), de velden CustomerID, RegisterID en ProductID spreken voor zich. De reden dat naast het aantal producten ook de totale prijs van wordt opgeslagen is omdat een product van prijs kan wijzigen. Door de totale prijs telkens in de database mee op te slaan is het mogelijk om zelfs na prijswijzigingen nog de exacte omzet van een avond te weten voor de statistieken.

Tot slot moet ook hier een Errorlog worden bijgehouden.

Tip

Bij verschillende tabellen moet er een datum en / of tijdstip opgeslagen worden. Hiervoor kan je het beste werken met een Unix timestamp (http://www.unixtimestamp.com/index.php)

Bij de klant moet een afbeelding opgeslagen worden. Hiervoor kan je BLOB storage gebruiken. De property mag van het type BitmapImage zijn. In de loop van de cursus komt er nog een oefening hoe je dit precies kan doen.

Todo

Maak een nieuw Portable Class Library project aan met als naam nmct.ba.cashlessproject.model en maak de nodige classes en properties aan zoals hierboven beschreven en zoals je voorzien hebt in je wireframes en user interface. De reden dat hier gekozen wordt voor een Portable Class Library is dat we later het model kunnen delen over verschillende projecten.

Schrijf de getters en setters voluit omdat het mogelijk is dat we deze later nog kunnen nodig hebben voor validatie.

Je hoeft je project **niet** in te dienen, maar zorg dat je het in **het labo van 13/10 tot 17/10** zeker bij je hebt om te kunnen voorleggen aan de labo docent.

Stap 4: connectie maken met een database

Bij deze stap zal je een database aanmaken voor je project en deze opvullen met de nodige testdata. Daarna kan je de helper class gebruiken uit het labo om de nodige selects en manipulaties te maken.

Todo: aanmaken database

Je kijkt zelf om je database aan te maken. Dit is omdat er redelijk wat verschillen zijn tussen de velden die vereist zijn. Sommigen kiezen er voor om extra data bij te houden en zo kan iedereen met een structuur werken die hem of haar het beste past. De database moet wel een MS SQL Server database zijn.

Tip voor aanmaken database

Neem een kijkje naar het model dat je aangemaakt hebt. Normaal gezien zal elke class een tabel worden en zo goed als elke property een kolom. Voor de ID's mag je overal autonummering gebruiken, uitgezonderd bij de medewerkers omdat deze hun ID moeten gebruiken om zich aan te melden aan een kassa.

Todo: data access code schrijven

Je zal ook de nodige methods moeten schrijven om data op te halen en opnieuw weg te schrijven. Je kan er voor kiezen om al deze methods op voorhand te schrijven of om ze on the fly aan te maken telkens je een bepaalde method nodig hebt. Maak in elk geval enkele methods om je connectie met de database te testen.

Voor deze stap maak je een nieuw web api project aan met als naam nmct.ba.cashlessproject.api.

Je hoeft je project **niet** in te dienen, maar zorg dat je het in **het labo van 17/11 tot 21/11** zeker bij je hebt om te kunnen voorleggen aan de labo docent.

Stap 5: gebruiken van MVVM

Tijdens de theorie heb je het concept van MVVM gezien en in het labo heb je een template gemaakt zodat je eenvoudig MVVM in je project kan gebruiken.

Todo: nieuw project aanmaken

De bedoeling van deze stap is om nu een nieuw project aan te maken op basis van deze template en al je reeds geschreven XAML code (en eventueel C# voor converters) in dit nieuwe project te plaatsen. Het nieuwe project krijgt als naam nmct.ba.cashlessproject.ui.

Maak ook een navigatie aan in je MainWindow zoals gezien tijdens de les en voorzie ook een ContentControl die als container voor je UserControls zal functioneren. Maak dan van elk Window een UserControl aan. Je kan meestal gewoon de code copy/pasten.

Test dan uit of je navigatie vlot werkt.

Todo: ViewModel

Maak dan in het ViewModel de nodige properties om de binding met de View te maken. Je kan bijvoorbeeld al een lijst van medewerkers, producten,... laten weergeven in de View. Dit moet nog niet volledig af zijn, maar probeer al zoveel mogelijk te doen. Dit zal je later werk besparen.

Indienen

Deze stap van het project moet ingediend worden op LEHO voor 29 November, 8:00. Je maakt een ZIP file van gans je project en dient dit in onder de naam **2NMCTGroepsNummer_Naam_Voornaam_BAP_01.zip** (bijvoorbeeld: 2NMCT1_Duchi_Frederik_BAP_01.zip).

Te laat of niet indienen resulteert in geen score voor deze tussenstap.

Stap 6: valideren van data

Het is belangrijk voor je project dat alle data die in de database komt gevalideerd wordt. Tijdens de theorie en het labo heb je hiervoor de nodige technieken gezien. Enkele voorbeelden zijn: bepaalde velden die verplicht zijn of een minimum aantal karakters vereisen. Verder moeten velden zoals een tijdstip of een e-mailadres aan een bepaald patroon voldoen.

Validatie is echter meer dan dat. Zorg er ook voor dat er bijvoorbeeld getest wordt wat de impact is van een record die verwijderd wordt. Dit kan je opvangen omdat je een error zal krijgen van ADO.NET indien dit niet correct afgehandeld is.

Todo: validatie toevoegen

Je kan het best de validatie uitvoeren op je model door gebruik de maken van data annotations. Je hebt deze techniek toegepast in het labo en er werd tijdens de theorie ook een demo gegeven die online staat. Zorg er echter ook voor dat controls zoals buttons niet enabled zijn op het moment dat het model niet valid is.

Stap 7: synchroniseren van de log files

Een van de requirements is dat de log files van elke kassa in de database van het IT bedrijf moeten komen. Er zijn verschillende manieren om dit te verwezenlijken. Je kan er zelf een stuk code voor schrijven en gebruik maken van een Windows Service, of je kan ook je SQL Server configureren dat dit automatisch wordt gedaan.

Todo: synchroniseren

Zoek zelfstandig naar een manier om dit probleem op de lossen. Je implementeert en test je oplossing. Je kan natuurlijk altijd je labo docent aanspreken om je hier bij te begeleiden.

Stap 8: testen, testen, testen

De laatste stap in je project is om het geheel grondig te testen. Probeer verschillende scenario's uit, schrijf waar nodig nog error handling statements (try... catch). Zorg er voor de de applicatie gemakkelijk te gebruiken is en foute input van de user correct opvangt. Test dus niet enkel de scenario's waar je de applicatie gebruikt zoals het hoort.

Een goede tip is om je project ook eens door iemand anders te laten testen. Externen zullen sneller fouten maken omdat ze de context van je applicatie niet direct kennen en meestal zijn ze ook kritischer dan dat je zelf voor je applicatie kan zijn.

Stap 10: indienen

Wanneer?

De deadline voor je project is maandag 5 januari 2015 om 8u. Het indienen van de opdracht na deze deadline kan resulteren in een gedeeltelijk of volledig verlies van punten.

Tijdens de kerstvakantie kan je nog mailen met vragen, maar hou er rekening mee dat docenten ook met vakantie zijn en een antwoord dus enige dagen op zich kan laten wachten.

Het is dus aan te raden om zoveel mogelijk van je project af te hebben zodat je tijdens het kerstverlof enkel maar nog kleine aanpassingen moet doen. Het feit dat de examens direct starten na het kerstverlof is nog een extra reden om voor het verlof zoveel mogelijk af te hebben.

Wat?

Je maakt 1 zip file die de volgende bestanden bevat:

- Als je broncode
- De log van Git
- Een bak file met je database

Voor de NMCT website en de infodagen is het ook altijd leuk om enkele screencasts te hebben van projecten. Wie wil mag nog een korte screencast (ongeveer 1 à 2 minuten) maken over zijn project en deze ook toevoegen aan de zip file. Je kan hier echter geen extra punten mee scoren, enkel eeuwige roem omdat je werk gebruikt wordt ter promotie van de opleiding.

Hoe?

De naam van de zip file is **2NMCTGroepsNummer_Naam_Voornaam_BAP_final.zip**. Je dient alles in op de dropbox van LEHO ter attentie van alle docenten. Controleer zeker nog eens bij je verstuurde items of alles ingediend is om misverstanden te voorkomen.