|  |  |
| --- | --- |
|  |  |
|  | Kevin Ly  Marco Havermans  City: Breda  Project group: 6  Employer: H.C.M. van Bueren  Date: 7/11/2014  Class: RIO4-APO2A/B  Version: 1.0 |
|  | Projectafbeelding |

# Organisatie

De applicatie van project Barroc-IT is ontwikkeld door team FourTwenty en de opdrachtgever H.C.M. van Bueren heeft ons daarvoor de opdracht gegeven. Voor meer informatie over de opdracht kunt u terecht bij een van de personen die een onderdeel zijn van de opdracht:

*H.C.M. van Bueren   
Kevin Ly   
Marco Havermans*

# De opdracht

**Barroc-IT**

We hebben de opdracht gekregen om een applicatie te ontwikkelen voor het bedrijf Barroc-IT. Het bedrijf is 10 jaar geleden gevestigd te Breda en heeft nu meer dan 100 man in dienst. De organisatie heeft verschillende afdelingen en ontwikkelt software voor zijn klanten. Zij willen dat wij een applicatie ontwikkelen, omdat ze problemen hebben met de communicatie tussen afdelingen. De gegevens worden niet goed doorgevoerd bij de afdelingen.

**Opdracht**

* Het is de bedoeling dat er een centraal klantenbestand komt om dit probleem op te lossen. Dit klantenbestand wordt bijgehouden door de afdeling financiën en alle andere afdelingen kunnen het alleen bekijken.
* Elke afdeling heeft een eigen centrale account om in te loggen.
* De klanten/facturen worden gearchiveerd indien iets wordt verwijderd. (geen permanente verwijdering)
* De afdelingen krijgen allemaal de standaard velden in de applicatie en die volgt:

|  |  |
| --- | --- |
| Veld | Omschrijving |
| Bedrijfsnaam (Company name) | Naam van het bedrijf (Name of the company) |
| Adres1 (Address1) | Het adres1 (The address1) |
| Postcode1 (Zip code1) | De postcode1 (The zip code1) |
| Woonplaats1 (City1) | De woonplaats1 (Cityname1) |
| Adres2 (Address2) | Het adres2 (The address2) |
| Postcode2 (Zip code2) | De postcode2 (The zip code2) |
| Woonplaats2 (City2) | De woonplaats2 (Cityname2) |
| Contactpersoon (Contact person) | De contactpersoon (Contact person) |
| Voorletters (Initials) | De voorletters (The Initials) |
| Telefoonnummer1 (Phone number1) | Het telefoonnummer1 ( Phone number1) |
| Telefoonnummer2 (Phone number2) | Het telefoonnummer2 ( Phone number2) |
| Faxnummer (Fax number) | Het faxnummer (the fax number) |
| E-mail (E-mail adress) | Het e-mail (the e-mail adress) |
|  |  |

**Afdeling Financiën**

* Financiën heeft de volgende velden nodig met omschrijving:

|  |  |
| --- | --- |
| Veld | Omschrijving |
| Bankrekeningnummer | IBAN Nummer |
| Saldo | De som van het aantal niet betaalde facturen. |
| Aantal facturen | Het aantal open facturen (niet betaald) |
| Omzet | De som van het aantal betaalde en niet betaalde facturen |
| Limiet | De maximale grens van het Saldo. |
| Grootboekingsnummer | Intern nummer |
| BTW Code | Het percentage van de BTW |
| BKR | Controleert of ze kunnen betalen. Ja/Nee |

* De afdeling financiën zou facturen moeten kunnen toevoegen, wijzigen en verwijderen. Het wijzigen mag alleen op typefouten en niet volledig worden gewijzigd.
* De afdeling moet kunnen zien welke klanten er nieuw zijn.
* Het moet mogelijk zijn om een lijst van debiteuren en crediteuren te bekijken.

**Afdeling Sales**

* + Sales heeft de volgende velden nodig: “Offer numbers, Offer status, Prospect Y/N, Date of action, Last contact date, Next action, Sale percentage, Creditworthy Y/N”. Ze moeten toegang hebben tot het klantenbestand.
  + Op het veld “Offer numbers” moet een begrenzing komen. De begrenzing moet standaard “1” zijn.
  + Als het veld “Prospect” op Yes staat betekent dat hij/zij een toekomstige klant wordt. Daardoor heeft hij/zij geen contract en is alleen zichtbaar voor de afdeling Sales.
  + Het moet mogelijk zijn om te zien of een klant een BKR melding heeft of niet. Deze melding komt van de afdeling Financiën.
  + Sales kan alleen maar klanten toevoegen en hun eigen velden bewerken.
  + De afdeling Sales wil de mogelijkheid hebben om een lijst van gegevens van de klant in te vullen. Ook willen ze dat het mogelijk is om afspraken in te plannen met de klant via de applicatie.
  + Alle gegevens moeten direct doorgevoerd worden naar de database vanuit de applicatie.
  + Er moeten foutmeldingen worden weergegeven als er een veld niet of verkeerd is ingevuld.

**Afdeling Development**

* + Development heeft de volgende velden nodig: “Maintenance contact, Open projects, Applications, Hardware, Software, Appointments, Internal contact person”. Ook moeten ze toegang hebben tot het klantenbestand.
  + Als Financiën meldt dat een klant niet betaald heeft voor een project, moet de afdeling Development dat meteen kunnen zien. Dan wordt dat project op inactief gezet.
  + De Development afdeling moet kunnen zien met welke projecten ze bezig zijn en of er onderhoud nodig is.
  + De applicatie moet de huidige applicaties, hardware en software van de klant weer kunnen geven.
  + Ook moet de afdeling Development kunnen registreren wanneer ze de klant voor het laatst hebben gesproken, waar ze over gepraat hebben en met wie ze hebben gesproken.

# De eisen

# Goal

Our employer give us an order to develop an application that will improve the communication between the departments of Finance, Sales and Development. He asked us to make a central database to use. In the first phase we have to present the following documents:  
- Mode of operation  
- Recap of the interviews  
- Renewed order  
- Prototypes of the schemes that are based on interviews  
- Quotation (offer)  
- Use-case diagrams  
- Use-case templates  
- Functional design  
- Technical design  
- Model dictionary  
- Data dictionary  
- Database design  
- Advice for optimization (Acceptation test)  
- Task partition development area

# Project orders

The project named Barroc-IT that is owned by B.C.M. van Bueren and will be executed by the developers group Team Fourtwenty.

|  |  |
| --- | --- |
| ***Team manager:*** | Kevin Ly |
| ***Secretary:*** | Pieter Hoek |
| ***Employee developer:*** | Marco Havermans |

## Barroc-IT

We get an order to develop an application for the company Barroc-IT. The company has over 100 man in their service and is settled in Breda 10 years ago. The organization has many different departments, it develops for their customers. They asked us to develop an application due the communication problems between the departments. The information between the departments will not always synchronize.

Order- One of the orders are to build a central database that will be managed by the finance department and all other departments can view the database information.  
- There is one central account to login each department  
- The files in the database will be archived instead of permanent deletion.  
- The departments get the standard fields in their application. See the table:

|  |  |
| --- | --- |
| Field | Description |
| Companyname | Name of the company |
| Address1 | The address1 |
| Zipcode1 | The zipcode1 |
| City1 | The city1 |
| Address2 | The address2 |
| Zipcode2 | The zipcode2 |
| City2 | The city2 |
| Contact person | The contact person |
| Initials | The Initials |
| Phonenumber1 | The phonenumber1 |
| Phonenumber2 | The phonenumber2 |
| Faxnumber | The fax number |
| E-mail | The e-mail address |
|  |  |

Finance department

* Finance has the following fields:

|  |  |
| --- | --- |
| Field | Description |
| Banknumber(IBAN) | IBAN account number |
| Saldo | The total of not-paid invoices |
| Aantal facturen | The count of not-paid invoices |
| Omzet | The count of not-paid and paid invoices |
| Limiet | The border of maximum balance |
| Grootboekingsnummer | Internal number |
| BTW Code | The percentage of the BTW |
| BKR | Control post for BKR alerts ( yes/no) |

* The finance department can add, modify or archive invoices. The invoices can only be modified for typos. It is not possible to edit the whole invoice.
* The department have to see which customers are old and new.
* There must be a list to view all debtors and creditors.

## Sales department

* Sales department has the following fields: “Offer numbers, Offer status, Prospect Y/N, Date of action, Last contact date, Next action, Sale percentage, Creditworthy Y/N”. Also they have access to the database.
* On the field "Offer numbers" there will be a border and this will be standard number "1".
* If the field "Prospect" is "yes" means that he or she is a future customer. He or she hasn't have a contract and is only visible for the sales department.
* It’s able to see a BKR alert of customers if they have it. This alert will be sent by the finance department.
* Sales department can only add a customer and modify their own fields.
* The sales department would like to have a list of information from the customers. Also they would like to plan a visit with the customers at the application.
* All kind of information must directly send to the database from the application.
* It must appears alerts if the fields are incorrect or blank.

## Development department

* + Development department have the following fields: “Maintenance contact, Open projects, Applications, Hardware, Software, Appointments, Internal contact person”. Also they must have access to the central database
  + If finance sent an alert that a customer didn't pay for a project, then the development department will get an alert to hold the project.
  + The development department would like that the application shows what kind of projects are running or they need maintenance.
  + The application have to show all current applications, hardware and software of the application.
  + Development department are able to register the customer and when they spoke with them. Also they are able to register which subject they spoken and with which customer.

# Project activities

For the current project we have to complete some activities.

Documentation:   
- Mode of operation  
- Plan   
- Renewed order  
- Task partition development area   
- Quotation (offer)   
- Use-case diagrams  
- Use-case templates   
- Class diagram  
- Activity diagram  
- Model dictionary  
- sequence diagrams

Preparing before build an application:   
- Prototypes of the schemes that are based on interviews  
- Technical design (Flowcharts)   
- Database design

Preparing after build of the application:   
- Create a functional-technical acceptation test  
- Executing of the acceptation test

Finished application:   
- Central database for use  
- Complete working application

# Project borders

The project Barroc-IT is a school project of the Radius College in Breda. A small group of developers will develop an application to improve the communication between the departments Finance, Sales and Development. The quotation will contain the costs.

Subjects what we add in the project:  
- We work with a plan  
- We have a group meeting every week  
- We hold a plan and we will update it if it's need  
- We will finish on our end date 7-11-2014

Subjects what we don't use in the project:   
- Don't add anything in the project that isn't agreed  
- We don't use any other program languages then C# and SQL

Other appointment during this project:  
- One user account each department  
- Application with maximum of three departments ( Finance, Development and Sales )

# Products

Documentation:   
- Plan   
- Renewed order  
- Task partition development area   
- Quotation (offer)   
- Use-case diagrams  
- Use-case templates   
- Class diagram  
- Activity diagram  
- Model dictionary  
- sequence diagrams

Mode of operation:   
- Goal  
- Plan  
- Project borders / Precondition  
- Products (Product of the project)   
- Quality  
- Risks

Weekly Reports

- Reports of meetings every week

Application  
- The application  
- The Database  
- The acceptation test

# Quality

In the first case we have to produce the documentation of the project and we will deliver the documentation before we start to develop the application. The surveillance of the quality will be done by everyone of the group and then we will hand it in.

After that we have to develop an application to improve the communication between the departments and the application have to be tested before the end date of the project. The project will have the following things:   
- Draw up of the basic program  
- Test of the basic program  
- Create a database  
- Link the application and the database  
- Test the application with the database  
- Bug fixing

To verify the application for use we make a acceptiontest for external users to let them try out the application. After that we will save up all the opinion and try to modify the application that is more user friendly.

On the last weeks we will fix errors if there is and this must be done one week before the end date. The project is ready to hand in if everything is done. If there is extra time we will try to make an Android application.

# Aanpak

**Planned tasks:**

**- Create a database** - Create tables (Marco Havermans)  
 - Create fields (Marco Havermans)  
 - Create primary and reference keys (Marco Havermans, Kevin Ly)  
 - Linking tables (Marco Havermans, Kevin Ly)  
 ***Check the database*** - Check tables (group)  
 - Check fields (group) (group)  
 - Check primary and reference keys (group)  
 - Check links between tables (group)

**- Create an application**  
 - Create a design of the prototypes (Marco Havermans, Kevin Ly)  
 *- Program codes*  
 *- Create classes (group)  
 - Program with code (group)  
 - Build the solution and include database (group)*  
 ***Check the application*** - Check the classes of the application (group)  
 - Check the program code (group)  
 - Check the working solution with database (group)

**- Create a acceptation test** - Create the acceptation test (Kevin Ly)  
 - Let external people test (Marco Havermans)  
 **Update the application with the acceptation test** - Optimize the application (group)  
 - Test the application so the application fully works (group)

**Completed tasks:**

**- Created database** - Created tables (Kevin Ly)  
 - Created fields (Kevin Ly)  
 - Created primary and reference keys(Marco Havermans and Kevin Ly)  
 - Linked tables (Marco Havermans and Kevin Ly)

***Checked the database*** - Checked tables (Marco Havermans and Kevin Ly)  
 - Checked fields (Marco Havermans and Kevin Ly)   
 - Checked primary and reference keys (Marco Havermans and Kevin Ly)  
 - Checked links between tables (Marco Havermans and Kevin Ly)

**- Created the application**  
 - Created the design of the prototypes (Marco Havermans and Kevin Ly)  
 *- Programmed*  
 *- Created classes (*Marco Havermans and Kevin Ly*)   
 - Built the solution and include database (*Marco Havermans and Kevin Ly*)*  
 ***Checked the application*** - Checked the classes of the application (Marco Havermans and Kevin Ly)  
 - Checked the program code for high-priority errors (Marco Havermans and Kevin Ly)  
 - Checked the working solution include the database (Marco Havermans Kevin Ly)

**- Created the acceptation test** - Created the acceptation test (group)  
 - Let external people test (group)  
 **Updated the application with the acceptation test** - Optimize the application (group)  
 - Test the application so the application fully works (group)

**- Checked the application works fine**  
 - Checked for input fields  
 - Checked for errors on click events  
 - Checked for user friendly  
 - Checked for lesser errors

Uitdraai uren

( Grafiek )

# Results and conclusions

Het resultaat van het project is dat we onze applicatie af hebben mits dat onze ex-teamlid tot spijt ons heeft verlaten vanwege privé redenen. Hierdoor hebben we de applicatie met tweeén af moeten maken en dit hebben we behaald.

In de vorige pagina is de informatie te vinden over het aantal gewerkte uren en de tussenresultaten. Ook is dit terug te vinden in onze centrale database die we online hebben onder de naam: Project\_Barroc-IT\_Groep6 op Github.

Vanwege dat we een teamlid missen zijn sommige documentaties niet afgerond wegens tijdgebrek. Dit wegens een onverwachte verlating van een van onze teamleden. Door deze situatie is er geen handleiding gemaakt voor de applicatie en is de voortgangsbewaking niet helemaal up-to-date.

Tot slot hebben we het doel wel behaald om een werkende applicatie te leveren die gevraagd was.