|  |  |
| --- | --- |
|  | **2014** |
|  | Team Fourtwenty  Kevin Ly  Pieter Hoek  Marco Havermans  City: Breda  Project group: 6  Employer: H.C.M. van Bueren  Date: 19/9/2014  Class: RIO4-APO2A/B  Version: 1.0 |
|  |  |

|  |
| --- |
| **[Barroc-IT]** |
| Mode of operation |

Table of Contents

[Chapter 1: Background 3](#_Toc398899908)

[Chapter 2: Goal 4](#_Toc398899909)

[Chapter 3: Project orders 5](#_Toc398899910)

[Barroc-IT 5](#_Toc398899911)

[Order 5](#_Toc398899912)

[Finance department 6](#_Toc398899913)

[Sales department 6](#_Toc398899914)

[Development department 7](#_Toc398899915)

[Chapter 4: Project activities 8](#_Toc398899916)

[Chapter 5: Project borders 9](#_Toc398899917)

[Chapter 6: Products 10](#_Toc398899918)

[Chapter 7: Quality 11](#_Toc398899919)

[Chapter 8: Projectorganisation 12](#_Toc398899920)

[Chapter 9: Planning 13](#_Toc398899921)

[Chapter 10: Costs and assets 14](#_Toc398899922)

[Chapter 11: Risks 15](#_Toc398899923)

# Chapter 1: Background

Team Fourtwenty is an application development group that develops different applications and it’s started in year 2014. This group has three workers and is developing software for Barroc-IT.

In the organization of Team Fourtwenty are many different functions. The following persons will work on the project:

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

Our group is known as one of the different developers of the Radius College in Breda. Our method of programming and subscribing is special so we have fewer rivals.

# Chapter 2: 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

# Chapter 3: 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.

# Chapter 4: 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

# Chapter 5: 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

# Chapter 6: 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

# Chapter 7: 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

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.

# Chapter 8: Project organization

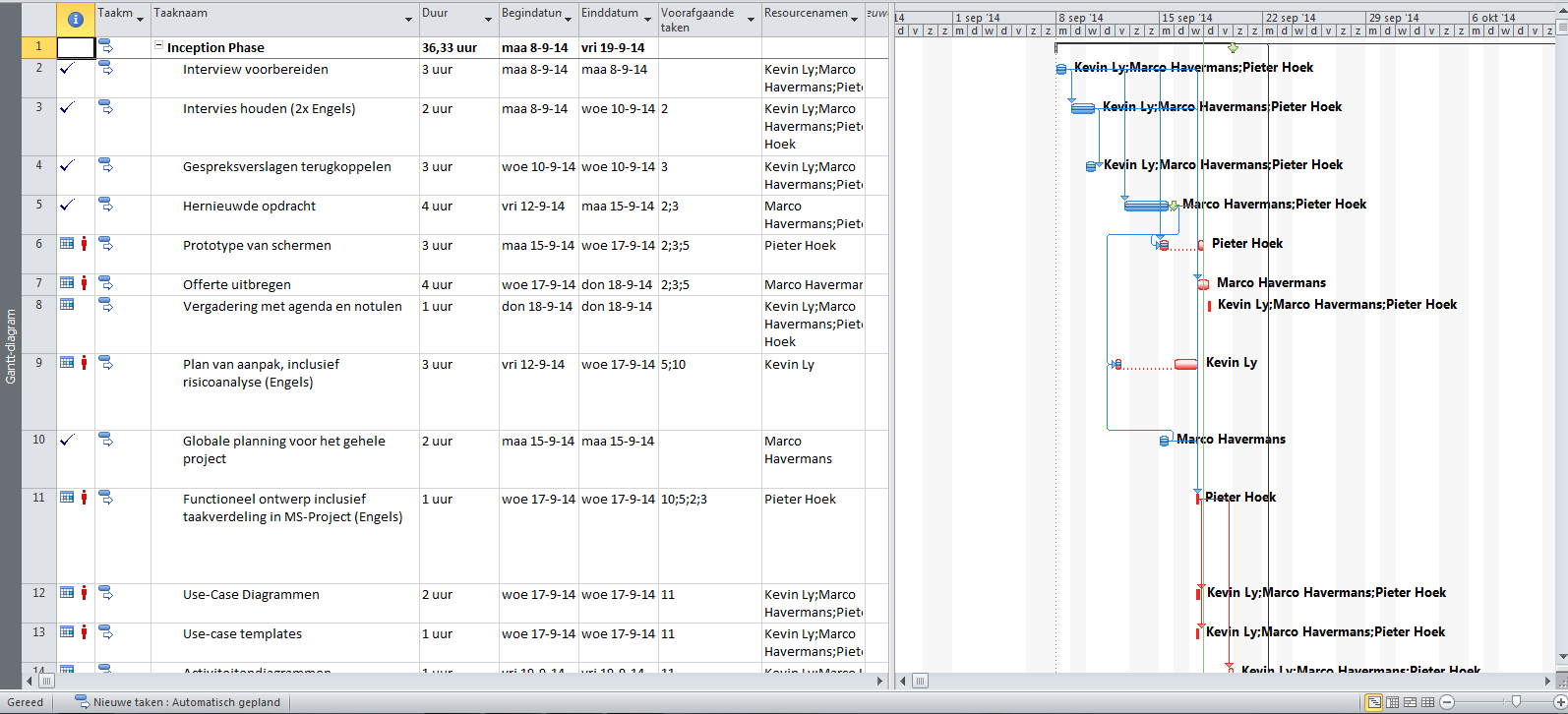
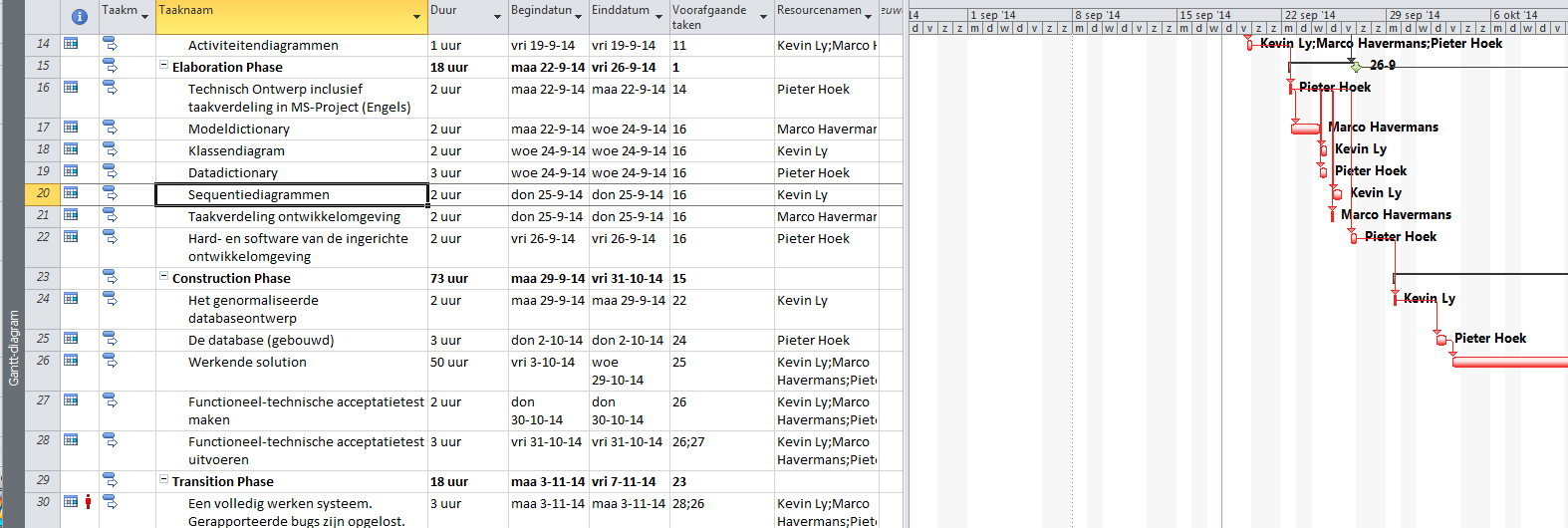
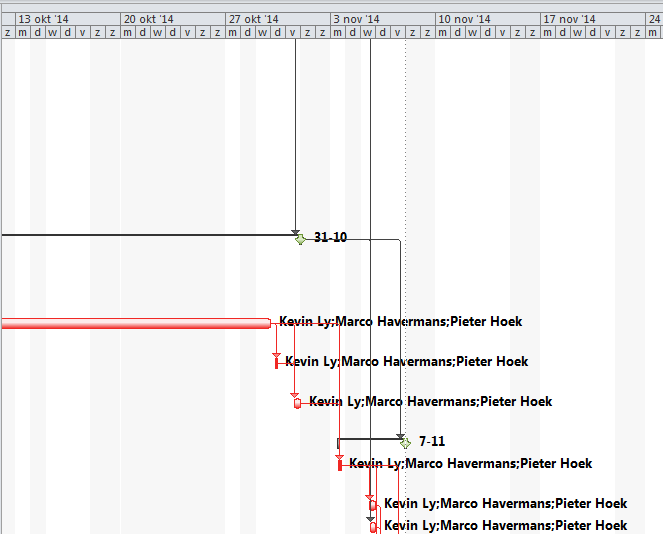
|  |  |
| --- | --- |
| ***Team manager:*** | Kevin Ly |
| ***Secretary:*** | Pieter Hoek |
| ***Employee developer:*** | Marco Havermans |

The employer of the project is always informed about the state of the current project. Reports have to be send to Mr. F. van Krimpen and are uploaded on GitHub. Weekly there is a group meeting to discuss about the state of the project. The schedule can be seen in the plan of action. The employer is informed about the reports that is made every week.



The communication between workers in the group will be on Skype, WhatsApp or Phone number. Workers also fill in how many they worked for the project. They will find it in the plan of MS-project. The meeting frequencies are weekly on Thursday.

# Chapter 9: Planning

# Chapter 10: Costs and assets

The project Barroc-IT is a free project of the Radius College in Breda. A small group of application developers will develop an application to improve the communication between the departments.

The next assets can be occur:   
- Workers will get sick (not available, appointments)   
- Bad project manager (bad leading to workers, bad communication with the employer, workers haven't enough information about the project)   
- Trigged because of social media or Internet  
- Bad communication between the workers in the group (too late inform if he or she is not available, running in late without to contact the workers, forget to ask for help on delay)

# Chapter 11: Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Description risk** | **Countermeasure** | **Chance** | **Consequence** | **Risk:  chance X consequence** | **Priority** |
| Interviews | Communicate with the team for further plans | 3 | 3 | 9 | High-priority |
| Create a renewed order | Take the team together to create | 2 | 3 | 6 | High-priority |
| Make prototypes from interviews | Communicate with team for lay-out or watch on internal website | 1 | 3 | 3 | Middle-priority |
| Create documentation (mode of operation, plan, quotation) | Try get more information how to create the documentation | 2 | 3 | 4 | High-priority |
| Make Use-case diagrams and templates | Focus the work and ask for support of team | 2 | 1 | 2 | Low-priority |
| Create functional design | Get more information about MS-project | 1 | 3 | 3 | Middle-priority |
| Create a activity diagram | Ask team for support | 1 | 1 | 1 | Low-priority |
| Create dictionary’s | Read information on MySite of the internal website | 1 | 1 | 1 | Low-priority |
| Create a technical design | Get information for MS-project | 2 | 2 | 4 | High-priority |
| Create classes and sequence diagrams | Search on Internet to get more information about the subject | 1 | 1 | 1 | Low-priority |
| Sharing the tasks for the development | Make a team meeting and share the tasks | 3 | 3 | 9 | High-priority |
| Create a place to develop soft- and hardware | Install programs what we going to use | 1 | 3 | 3 | Middle-priority |
| Create a design for the database | Ask team for support if there are problems | 3 | 3 | 9 | High-priority |
| Build the database | Communicate with the team if there are errors that can't be fixed by the person working on it. | 3 | 3 | 9 | High-priority |
| Build the application for the departments | Ask support if there isn’t enough time | 3 | 3 | 9 | High-priority |
| Create a acceptation test | See documentation for how to make a acceptation test | 2 | 2 | 4 | Middle-priority |
| Use the acceptation test | Ask people to try the test, write the negative things on paper | 1 | 3 | 3 | Middle-priority |
| Hand in the application | Check for bugs beforehand in! | 3 | 3 | 9 | High-priority |
| Write a user manual for the application | Check for typo beforehand in to users | 3 | 3 | 9 | High-priority |
| Prepare for presentation | Ask team for support to create a presentation | 2 | 2 | 4 | High-priority |