Hernieuwde opdracht

At this moment in time all the data is kept in word and excel files, and the departments we interviewed want that changed.  
The communication between the departments isn’t working as efficient as it could be because of this system.

Barroc-IT grew fast in a short amount of time and couldn’t really keep up with all the work they needed to do, that is why we were chosen to make this product.

After the interviews, we agreed on a few requirements for the final product.

First of all: Who can do what in the application?

* The sales department has to see if a customer is credit worthy, and they want to see when a customer has a payment delay.

In the database the sales department wants to see and edit the following data:

* Bank account number
* Credit balance
* Number of invoices
* Gross revenue
* Limit
* Ledger account number
* Tax code
* The chief executive said in the interview that he isn’t really going to use the application, but it will probably come in handy to give him access to all the data the other departments have access to.
* The finance department has to be able to access all the bills and payments. They want to see if a customer is credit worthty.

They also want to be able to edit:

* The numbers
* Company name
* Address
* Zipcode
* Place
* Contact person
* Initials
* Phone number
* Fax number
* E-mail
* The development department wants to have access to the maintenance of the application, and wants to see the status it is in.

They also want to see and edit the following data in the database:

* Company name
* Address
* House number
* Zip code
* Residence
* Contact person
* Telephone number
* Fax number
* E-mail

The data they only want to see is:

* Maintenance contract
* Open projects
* Applications
* Hardware
* Operating system
* Appointments
* Internal contact person

The application’s layout needs to follow Barroc-IT’s own style that is displayed on their website. The application itself can just have a simple compact layout, because it would probably be more clear to understand it like that.

**There are also a few requirements for the system:**

* It needs to choose a customer
* Add a bill
* Select payed bills
* Deactivate customers(not deleting them)
* The system needs to calculate if a customer is over their limit
* If the bill isn’t paid in time, the application needs to stop their current project
* It needs to display a message to contact sales when this happens
* When the bill is paid, the customer gets deactivated at sales and activated at development.

**There are also some could haves for the system:**

* A search function
* A filter function

The help function needs to explain what all the buttons do.