Version: 1.1
Datum: 2015-05-13

evry.com

Assignment

eCommerce (Magento)

Technical speficification

1 Backgrund

Nolia want to create a good user experience for their customers. The problem today is that customers call directly to the Nolia instead of using the Web to shop. The mission is to get their clients to start the e-shop on-line.

The eCommerce site should be able to have two languages (swedish, english) but only one currency: Swedish crowns.

2 Business Goals

- Easier for customers to rent products
- Increased cross-selling
- Connection to CRM system
- Increase the number of reservations via the web
- Sell tickets

3 Audiences

Exhibitors (hiring at the fair)
External organizers
- individuals / companies to the event
- Organizers of the event want to rent
- Competitors who want to rent

Individuals who want to buy a ticket to the fair

4 User Scenarios

4.1 Exhibitors

1) All new customers who are exhibitors call Nolia and say they want to be able to e-shop.

2) An email is created from the CRM with a link to the user who links the user to the correct exhibition and includes the customer's reservation number. When the user clicks on the link, they can start shopping.

Ex. the URL sent in the email:
http://shop.nolia.se/monterservice/stora-nolia-massan/

3) The user can choose the products that can be hired for this particular exhibition.

4) The Order is placed (sent to CRM)

5) The order is created in CRM.

6) The exhibitor will receive a reminder e-mail if they have not booked through the booth service yet. Eventual new prices. Link to the right. This message comes from the CRM.

4.2 People who want to rent products

1) A person from a company enters http://shop.nolia.se/

2) The visitor clicks through to rental of products.
http://shop.nolia.se/products/

3) The visitor selects city , for example . Pitea, all products belonging Pitea appear .
http://shop.nolia.se/products/pitea/

4) The visitor add products to the shopping cart
http://shop.nolia.se/products/pitea/

5) The visitor enters a specific product and read more about the product.
http://shop.nolia.se/products/pitea/lamp/

6) The visitor add the product to the basket:
http://shop.nolia.se/products/pitea/lamp/

7) The visitor press basket and come to CHECKOUTPAGE .

8) They choose what time they want to hire .

9) Select whether to pick themselves or get it delivered .

10) If you want to get it delivered should be able to choose a time for delivery .

11) In confirmation of the reservation states that the reservation is not approved until they get an ok from the Nolia . Need to be reconciled manually.

4.3 Visitors

1) A person from a company enters http://shop.nolia.se/

2) clicks through to I want to buy a ticket . " " Tickets for our trade fairs and events " "

http://shop.nolia.se/massor/

3) The visitor selects evenamang , as large Noliamässan

http://shop.nolia.se/massor/stora-nolia-massan/

4) The visitor add tickets to the basket

http://shop.nolia.se/massor/stora-nolia-massan/

5) The visitor add tickets to the basket fill in their email and buy ticket

4) Visitors can enter their card information (DIBS)

5) It generates an email to the visitor with a QR code from CRM.

5 Integration Flow

5.1 Campaigns

Before integrating products we need to integrate all campaigns (fair, area) for all three audiences from CRM. These top items is listing in the ContentPage for all areas

1) Exhibitors
    a. Fair (image, description, link to the productlisting page)
2) Visitors
    a. Fair (image, description, link to the productlisting page)
3) People who would like to rent
    a. Areas (image, description, link to the productlisting page)

http://www.nolia.se/API/GetCampaigns/Exhibitors

{

 results: [{

  topItem: { name: "Stora Nolia massan", id: “Campaign-Ex-1”, description: "Welcome to our biggest fair in scandinavia." },

  topItem: { name: "Skogs mässan", id: “Campaign-Ex-2”, description: "Welcome to our biggest fair in forrest area." }, }]

}

http://www.nolia.se/API/GetCampaigns/Visitors

{

 results: [{

  topItem: { name: "Stora Nolia mässan", id: “Campaign-Vis-1”, description: "Welcome to our biggest fair in scandinavia." },

  topItem: { name: "Skogs mässan", id: “Campaign-Vis-2”, description: "Welcome to our biggest fair in forrest area." },

 }]

}

http://www.nolia.se/API/GetCampaigns/Rents

{

 results: [{

  topItem: { name: "Piteå", id: “Campaign-RentArea-1”, description: "Have a look at our range of products in Piteå." },

topItem: { name: "Umeå", id: “Campaign-RentArea-2”, description: "Have a look at our range of products in Umeå." },

}]

}

5.2 Products

GetProducts () - scheduled job, magento asking about CRM data.

http://www.nolia.se/API/GetProducts/

{

results: [{

product: { productNr: "Prod-12345", name: “Lamp 1”, description: "The best Lamp we have!" },

product: { productNr: " Prod-12322", name: “Lamp 2”, description: “One of our best lamp” }

}]

}

5.3 Prices

GetPrices () - scheduled job, magento asking about CRM data.

The campaigns have different products in their range. If the product is associated to a price for a campaign it would be displayed in the CampaignPage. For example:

http://shop.nolia.se/products/Umea/ have only one product in their range (as example below).

http://www.nolia.se/API/GetPrices/

{

results: [{

campaign: { id: " Campaign-Vis-1 Campaign-RentArea-1", [{

product: “Prod-12345”, price: 1200 },{

product: “Prod-12322”, price: 1100 }

}]},

campaign: { id: " Campaign-Vis-1 Campaign-RentArea-2", [{

product: “Prod-12345”, price: 1000 }

}]},

.....

}]

}

5.4 Check Out Flow

All these flows are triggered from Magento and sends the information to the CRM.

1) Exhibitors (sending information through the API to CRM)

- Reservation number
- Stand number (hidden field)
- Business
- Email Address
- Contact Stand
- Phone Personal Stand
- Other
  Billing address
  - first name
  - surname
  - Company
  - Org.number
  - address
  - zip code
  - city
  - Adress2
  - Ref Invoice

2) Rental (sending information through the API to CRM)

- FromDate
- ToDate
- Delivery (0,1,2)
- FirstName
- SurName
- Phone
- DeliveryAdress1
- DeliveryAdress2
- Other

3) Buy the ticket (sending information through the API to CRM)

o Email Address

o Payment status (0 = Not Paid, 1 = Paid), managed based on DIBS.

Orders are stored in the CRM. The invoice is created in the CRM.
CRM offers different web services. Each request has a webservices. All requests made via REST API.

6 Business Rule

It is only possible to add products and buy from ONE campaign at the time.

- If the user try to add products from another campaign the user will get a warning. Alert: “You are not allowed to add products from other campaign.”

7 Interface / CMS: Magento eCommerce

Magento 1.9.1.
Magento the interface. Where the customer surfs in and shop online. The editor creates and edits the page content using CMS tool available in Magento.

If the user logs in to the site, there is a authentication against CRM.

7.1 Page Types

• StartPage
• CampaignePage
• ProductListingPage
• Product Page

CheckoutPage (Magento Standard)

8 Techniques

• Code: HTML5 / CSS3 / Sass / Javascript / PHP
• Operating System: Linux or Windows
• Database: MySQL (Magento)
• Server: Apache 2.x or Nginx 1.7.x
• Magento 1.9.1
• Library: jQuery, prototype.js, slideshow.js, modernizr, selectivzr, datepicker.js
• Front-end tools: Shallow
• Front-end test: Jasmine / PhantomJS

9 API

All APIs will be REST API.

10 Plugins

• Magento: English pack
• JavaScript: Slideshow, datepicker

11 10 3-party applications

• Dibs (Payment Solution)

12 Languages

• English / Swedish - In interface
• English - Magento admin

13 Browsers

- Chrome
- Firefox
- Internet Explorer> = 10
- Smartphone (iOS)
- iPad (iOS)

14 Sitemap

- StartPage (always the same homepage for all audiences)
  - CampaignPage
    - ProductListingPage
      - Product Page
  - CampaignPage
    - ProductListingPage
      - Product Page

13.1 Exhibitors

Home Page: http://shop.nolia.se/

CampaignPage: http://shop.nolia.se/monterservice/

ProductListingPage: http://shop.nolia.se/monterservice/{MässansNamn}/

Example:

http://shop.nolia.se/monterservice/stora-nolia-massan/

http://shop.nolia.se/monterservice/skogsmassan/

http://shop.nolia.se/monterservice/badmassan/

Product Page: http://shop.nolia.se/monterservice/{MässansNamn}/{ProductName}/

Example:

http://shop.nolia.se/monterservice/stora-nolia-massan/Lampa/

http://shop.nolia.se/monterservice/stora-nolia-massan/Barbord/

http://shop.nolia.se/monterservice/stora-nolia-massan/Matta/

CheckOutPage

In case an exhibitor will receive an email from the CRM so the user will get for example: http://shop.nolia.se/monterservice/stora-nolia-massan/

14.1 Visitors

StartPage: http://shop.nolia.se/

CampaignePage: http://shop.nolia.se/massor/

ProductListingPage: http://shop.nolia.se/massor/{mässa}/

Example

http://shop.nolia.se/massor/stora-nolia-massan/

CheckOutPage

14.2 Persons wishing to rent

StartPage: http://shop.nolia.se/

CampaignePage: http://shop.nolia.se/products/

ProductListingPage: http://shop.nolia.se/products/{Ort}/

Example

http://shop.nolia.se/products/Umea/

http://shop.nolia.se/products/Pitea/

Product Page: http://shop.nolia.se/products/{Ort}/{ProductName}/

Example

http://shop.nolia.se/products/Pitea/Lampa/

http://shop.nolia.se/products/Umea/Barbord/

CheckOutPage

14 Architecture