NCRM- SOFTWARE SPECIFICATIONS DOCUMENT

June 10th, 2017

ABSTRACT

An SSD for NCRM for Zetes Netherlands

Tariq Hussain

Marketing

DOCUMENT APPROVAL

Project Sponsor & Company Tutor- Saskia Eijkelhof	Project Officer- Tariq Hussain
Date **Signature**	**Date** **Signature**

DOCUMENT CONTROL

Document Title	NCRM- Software Specifications Document
Version No	v1.0
File Name	NCRM- Software Specifications Document _v1.0
Date:	June 10th 2017
Distribution	NCRM- Software Specifications Document_v1.0.doc

DOCUMENT MODIFICATION HISTORY

Version Number	Primary Author	Description of Version	Date Completed
0.1	Tariq Hussain	Draft version	3rd April, 2017
0.2	Tariq Hussain	Draft version	29th April, 2017
0.3	Tariq Hussain	Draft version	18th May, 2017
0.4	Tariq Hussain	Draft version	31st May, 2017
1.0	Tariq Hussain	Final version	10th June, 2017

TABLE OF CONTENTS

1. INTRODUCTION

1.1 PURPOSE OF DOCUMENT

This document will provide the necessary conceptual foundation required to start the building of a web-based application, which will be a companion app for the Salesforce production environment here in Zetes Netherlands. The document will provide the necessary software specification requirements as envisioned by the client with sufficient detail. Further, the document will serve as objective evidence for the development team when implementing the functionality and enforcing commitment to the scope of the project. This document is to be treated as a very open-ended and non-technical document, which serves the development team to capture exactly what the client envisions before it forms a proper software and its Design and Specification. It can surely be used to further develop into a fully compliant of all SSD standards or be made anew.

1.2 SCOPE AND BACKGROUND

Here in Zetes Netherland, the Sales and Marketing departments must manage their customer base efficiently. The CRM (customer relationship management) software they use is Salesforce; however, there is not one person who is assigned the role of one full time administrator. Although, proper Salesforce administration is quite beneficial for an efficient CRM experience it is however not seen as a core-function in the company. Because of no one available for solely maintaining the data in Zetes Salesforce, not to mention the tedious nature of the task, the quality of the data slowly degrades and it falls to on premise software development to smoothen the process of putting the information into Salesforce easy and more importantly time and cost effective.

As a research project, there was the task to develop a contingency plan in case there was not a solution in the market to counter this unique problem faced here. A decision was taken to conceptualize this idea of a minimalistic CRM application from a business requirement standpoint only. Developing a Salesforce application requires experience in APEX i.e. the language used in the Force.com developing platform on top of experience with developing software in general.

The “NCRM” web application is built using Salesforce but is tailored to meet the exact needs of the Sales and Marketing teams. It will have a simple User Interface with an emphasis on the use of a form layout for each category. It differs from the Salesforce Classic interface in that it does not employ fields, which are not required by Zetes. It is envisioned to be minimalistic, functional and presentable in nature to be exposed to customers and clients, even though it is an internal app. Every user will be able to login to this website using their Salesforce user-name and password. The site can be developed using Force.com-the Salesforce web development platform for custom websites, as this will ensure tighter integration. The app will also produce the possibility to produce a .csv file which can be uploaded to Salesforce or simply populate all the fields as a primary function. This application will also prompt the user to use all the Salesforce Best Practices for data-entry provided by Zetes while focusing only on the mandatory fields.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

TERM	DEFINITION
CRM	Customer Relationship Management
Admin	Admin is a User with certain additional controls and access to features.
User	Sales and Marketing persons in Zetes Netherland
Contact	Any individual that can add value to business Zetes.
Company	Any entity that engages in business with Zetes.
Account	Sales and Marketing term for company and which will be used in Salesforce Zetes.

Contact Type	A classification among contacts based on relevance.
Business Unit	A logical element or segment of a company representing a specific business function.
Session	The duration of time within which a User adds new Accounts or Contacts for a specific Marketing Campaign in Salesforce.
General Prospection	All the Accounts and Leads added which do not fall under a Marketing Campaign.
Element	A holder of a unique functionality.
Overtime	Extra units of time in days given to a User for a Session.

2. OVERALL DESCRIPTION

This section will describe the functionalities of the entire system and how the different parts of the system interacts with each other to fulfill its CRM objectives and in the context of the stakeholders, the stakeholders being the Sales and Marketing personnel of Zetes Netherlands.

2.1 PRODUCT PERSPECTIVE

NCRM is a web-based application platform that integrates with Salesforce specifically designed for the Sales and Marketing department of Zetes Netherlands. It will allow the Admins to interact with the Users and vice versa in a controlled approach to make the task of loading data into Salesforce seamless and user-friendly. It will allow Admins to facilitate the Users effectively without compromising the ease of use. To manage Users, Admins will have additional rights and will hold that single position in their respective country division, as only one Admin per division will be necessary. However, an Admin can declare their rights to another User and switch back to being a User in their respective country division. For those purposes a database will also be established where Admin and User information can be stored.

The data entry system works by submitting forms where each field is mapped to an object in Salesforce production. It will be crucial to enforce the rule that entries with pending fields do not get uploaded to Salesforce, so as to promote quality information. Upon fulfillment of such form, the platform offers the option to Save and directly export to Salesforce or store it for a certain period of time depending on the type of Session. This brings up the use of Sessions and Services here. In a Session there will be a set time-frame and will be associated to a particular Salesforce campaign which also is bound by a time-frame. However, a Service is not time-bound and only applicable for the Users who are not associated with a particular Session. The Admin will create a Session for Users during which the data needs to be filled into the platform. The Admin will set a particular time-frame during which the session will run, and also associate particular Users to that Session. If a User requires more time to fulfill their data-entries properly, the User will request for additional time within the time-frame. The Admin will receive a notification immediately regarding a User's request for additional time to approve or disapprove. Upon approval that User alone receives the additional time to remain in that Session and fulfill all data entries. After a Session ends all the pending fields are reported to the Users for fulfillment after which they are loaded to Salesforce, however those entries are now transferred to the Permanent Session which is associated to global-campaign created for NCRM in Salesforce.

In order to fulfill quality data-entry practices, data-entry conventions will be enforced. User conformity coupled with an optimized selection of information will ensure a clean and uniform data throughout salesforce production.

2.2 PRODUCT FUNCTIONS

This web-application shall perform two primary functions. First it will serve as a very simplified CRM that is tailored to the Sales department of Zetes Netherlands and secondly it will export information to Salesforce.

To support these primary functions other supporting functions are in place, which are mentioned below:

- Updating, Inserting, Editing Contacts & Accounts.
- Only those Exports that have reached completion enter Salesforce.

2.3 USER CHARACTERISTICS

The users that will interact with this application system will be from the Sales and marketing department of Zetes Netherlands. They need to interact with customers and leads and can easily have access to their information. They have Salesforce accounts and have experience using a CRM software service, however they are not at ease with the current system. The focus on an organized UI with the least amount of repetitive work has been emphasized. It is company policy to use and develop Salesforce as their primary CRM software, hence they are open to any service application that will integrate with Salesforce.

2.4 CONSTRAINTS

The constraints to this project can be categorized into two broad stations. One station which is Salesforce as a development platform itself and the other is finding the development team with experience. To provide the information pertaining to the constraints faced using the Salesforce platform cannot be justified by a sub-section in this document as it is quite a significant topic and it would de-rail from the purpose in this document It is better off as a standalone with ample referencing from the vast information library of Salesforce.

However, some critical constraints that does align immediately with the conceptual understanding of NCRM would have a mention here. Since the back-end of this application will integrate with Salesforce we have a few constraints to consider when it comes to the edition in use i.e. Enterprise. Salesforce provides a native infrastructure called Force.com which uses a programming language called APEX that can be used to make custom apps. Apps developed using Java will need to integrate with Force.com to meet specifications. The choice of the protocol to be used lies on the developer based on an educated judgement depending on the requirements presented here and other constraints. Here is a list of API provided by Force.com which employ their respective protocols, data format and communication. [1]

API Name	Protocol	Data Format	Communication
REST API	REST	JSON, XML	Synchronous
SOAP API	SOAP(WSDL)	XML	Synchronous
Bulk API	REST	CSV, JSON, XML	Asynchronous
Metadata API	SOAP (WSDL)	XML	Asynchronous
Streaming API	Bayeux	JSON	Asynchronous (stream of data)
Apex REST API	SOAP(WSDL)	XML	Synchronous
Apex SOAP API	SOAP(WSDL)	XML	Synchronous
Tooling API	RESR or SOAP(WSDL)	JSON, XML, Custom	Synchronous

2.5 ASSUMPTIONS AND DEPENDENCIES

ASSUMPTIONS:

- Exports are always new and potential-duplicates are not exported.
- All fields of entry in the Submission forms including Custom Fields are mandatory, hence fields that meet the business requirements in exact by the Zetes Salesforce production should be created.
- Exports by Users with the right privileges are validated from reliable sources.

DEPENDENCIES:

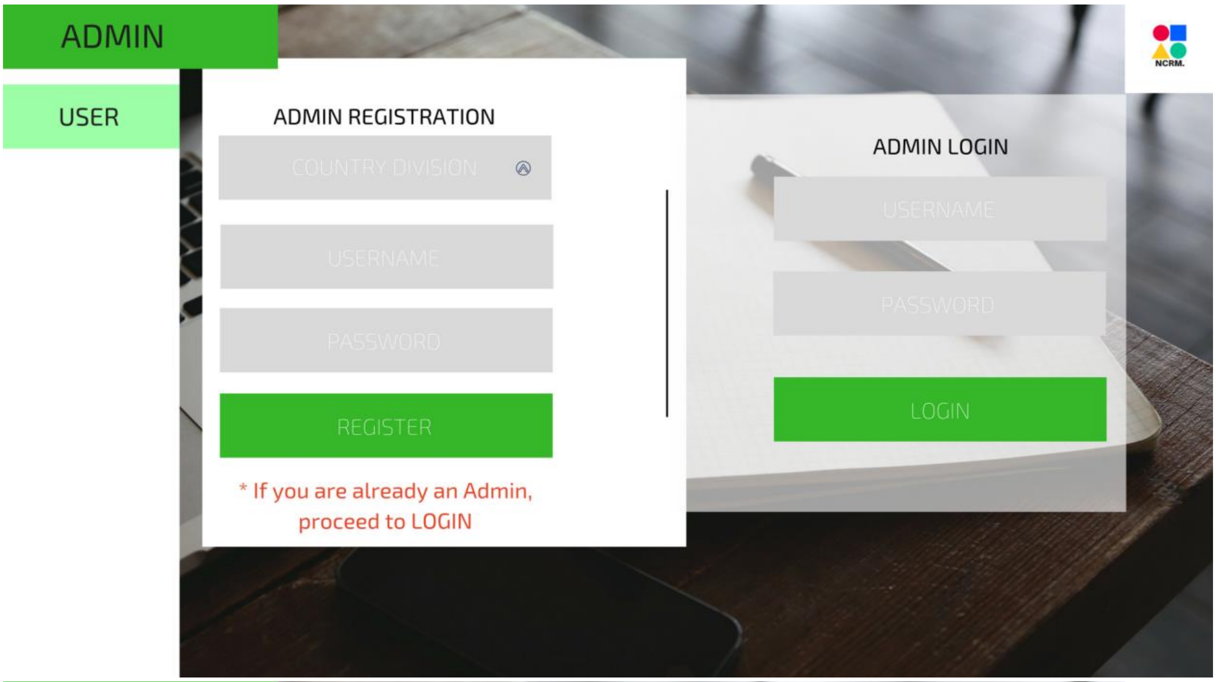
- A User has a Salesforce license provided by Zetes.

3. SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

- **ADMIN & USER-** This page has a side navigation bar with two buttons, ADMIN and USER. Clicking ADMIN button brings out two forms: Admin Registration and Admin Login. The Admin Registration form has the following fields: Username, Password, Country Division (dropdown menu list: Corporate, Austria, Belgium, France, Germany, Netherlands, Switzerland, Czech Republic, Denmark, Ireland, United Kingdom, Israel, Greece, Italy, Spain, Portugal, South Africa) and a "REGISTER" button. The Admin Login form has the following fields: Username, Password and a LOGIN button, as shown in Fig.1.



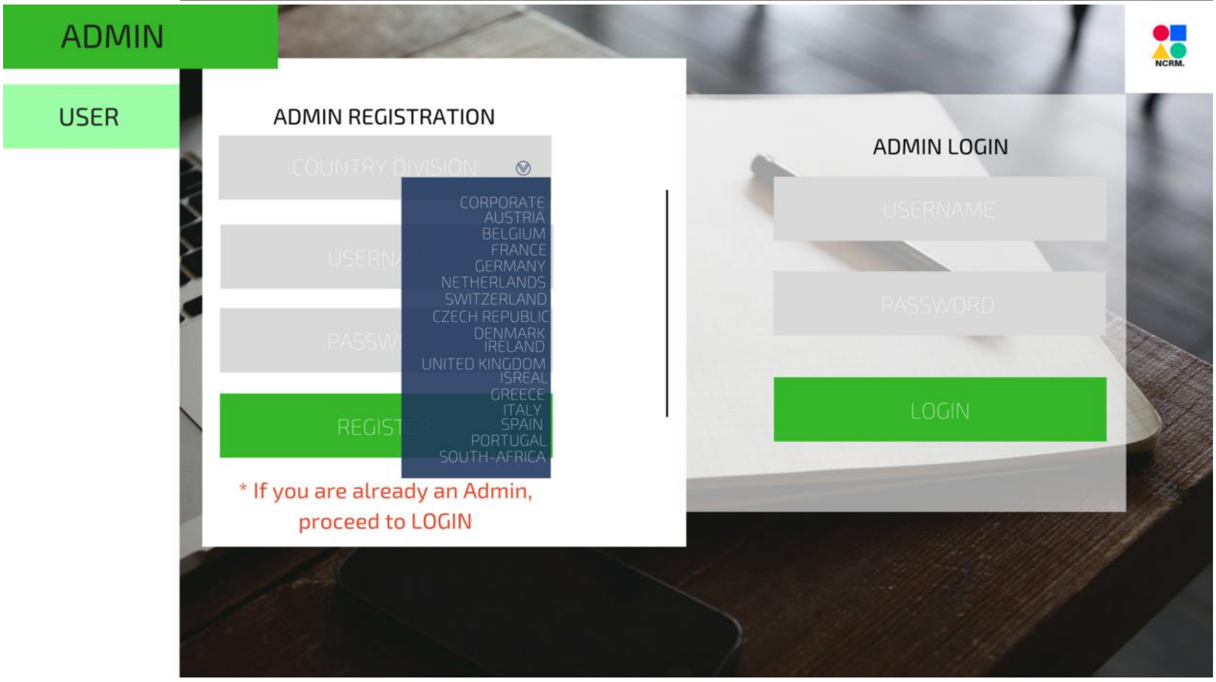


Fig.1

Whereas clicking USER button brings out a form with field Username and a LOGIN button as shown in Fig.2.

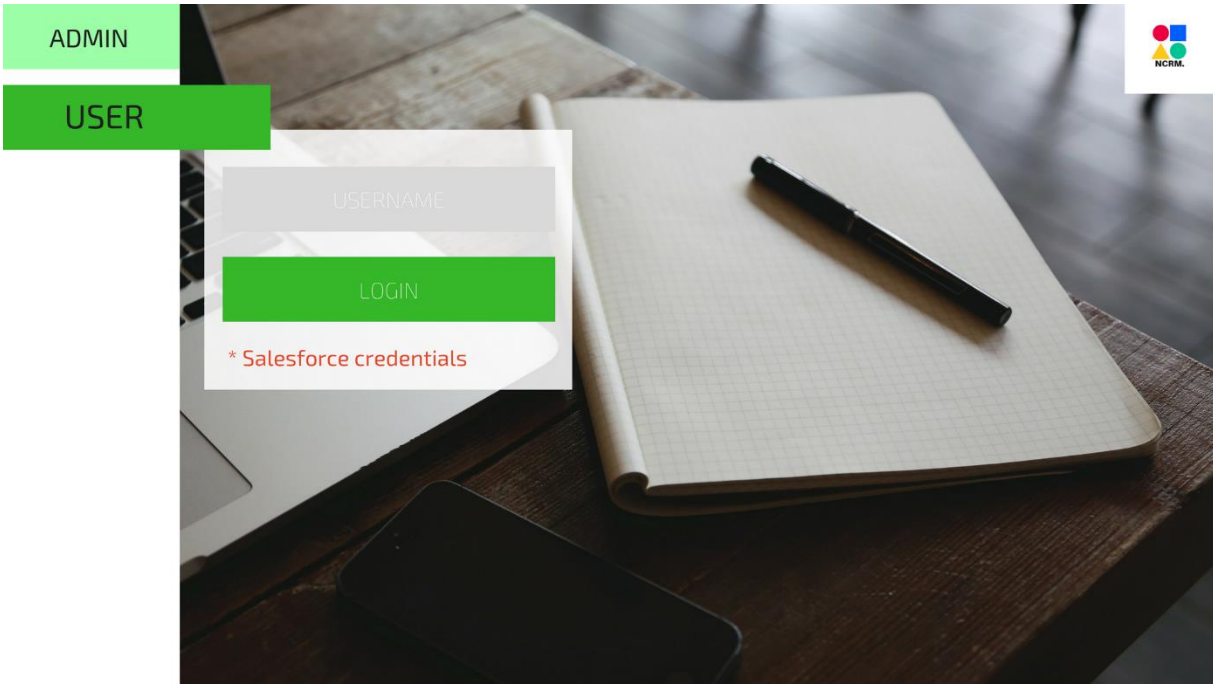


Fig.2

- **HOME**- This page shows the User information i.e. Full Name, Job Title, Department and Session information if the User is associated with a Session. A Session shows the following fields: Session Name, Start-End Dates, Associated Users, and a START/STOP button, see Fig.3.

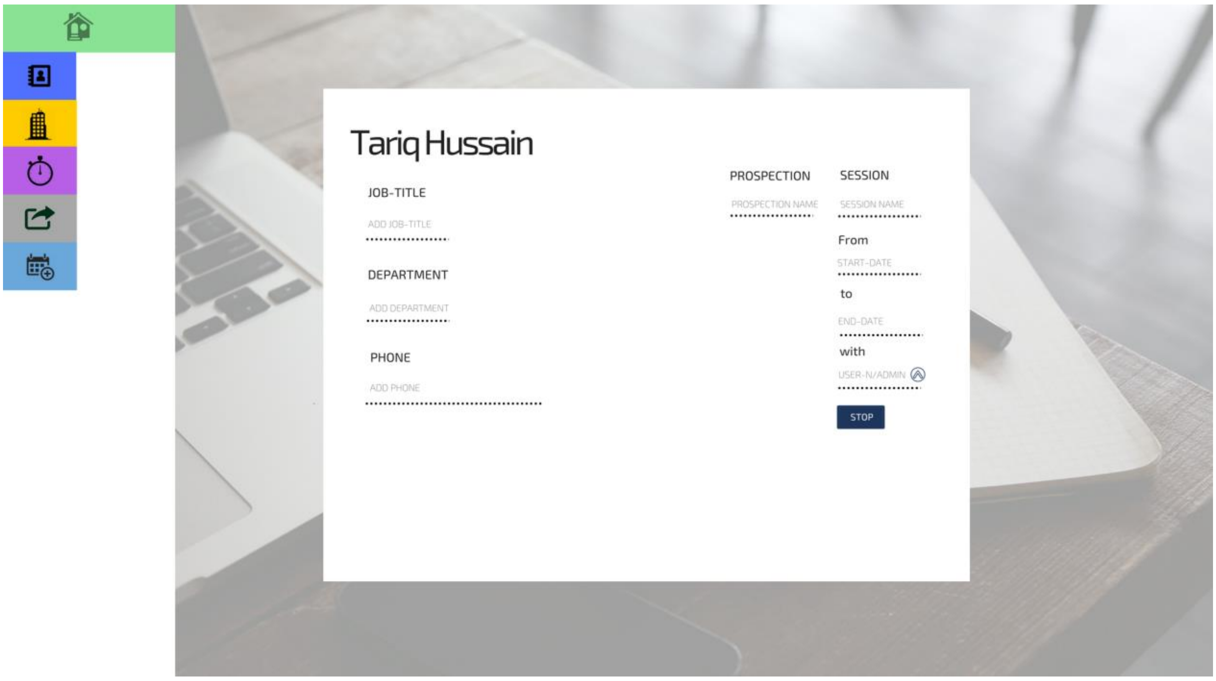


Fig.3a

The view for the Admin for this page has a drop-down icon under prospection as pointed out by the red arrow below in Fig.3b.

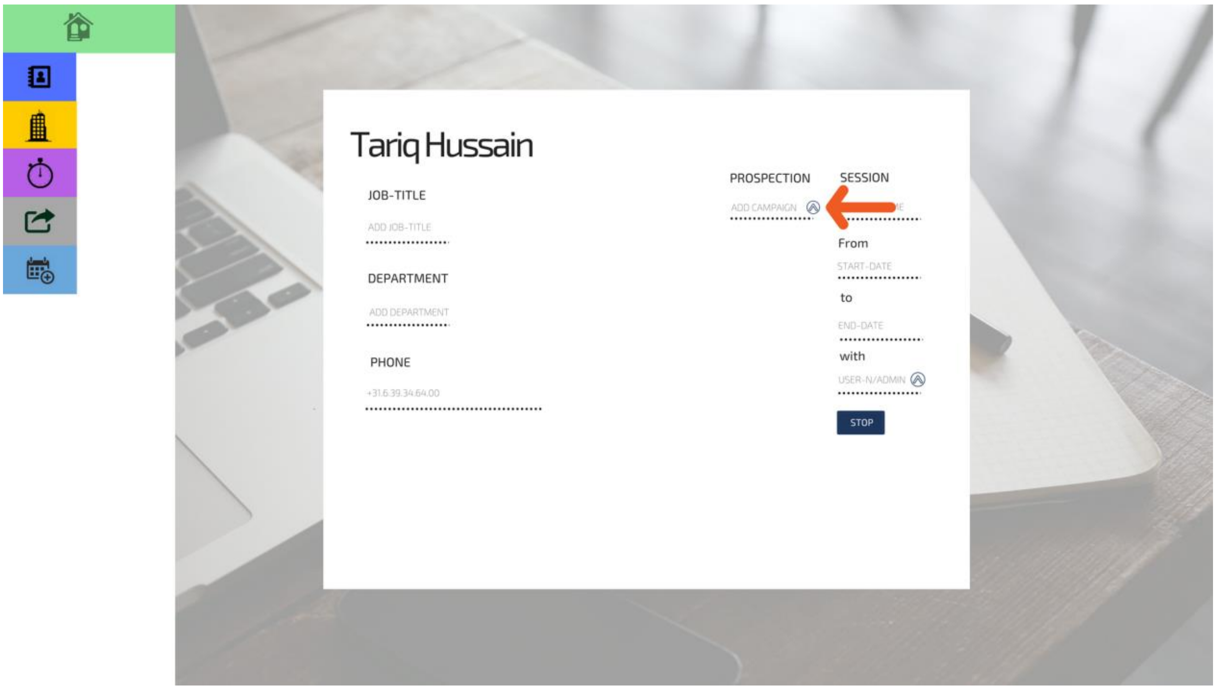


Fig.3b

- **CONTACTS**- This page shows a table which holds the default column headers or other custom fields made by the user for each lead. The following fields are however fixed to view: Checkbox, Full-Name, Company, Level, see Fig.4.

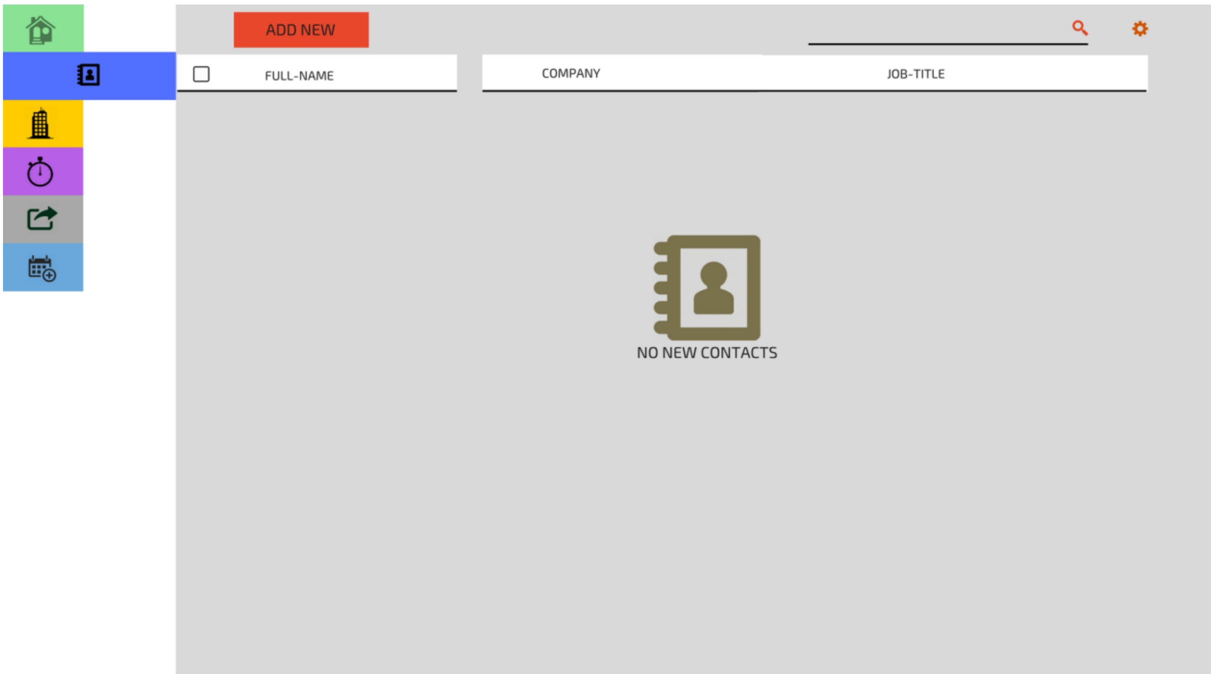
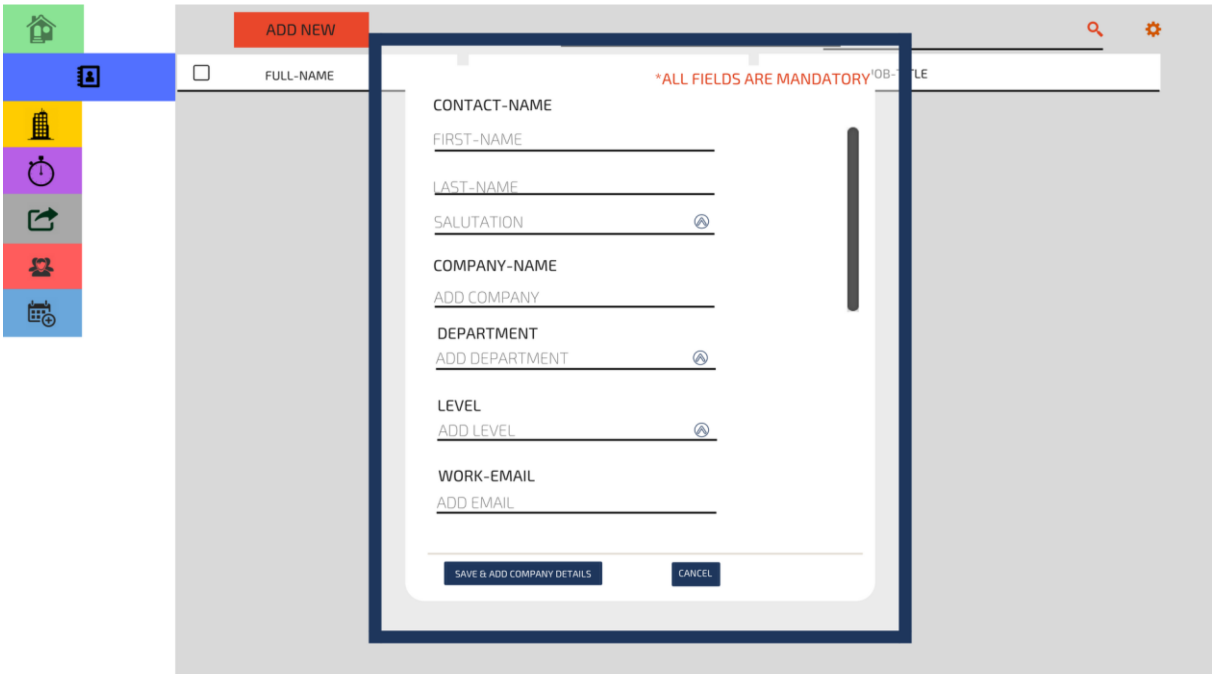


Fig.4

The horizontal menu bar above this table has two states, i.e. before and after records are on Checkbox. Before Checkbox State shows the following elements:

"Add New" button brings out a "Add New Lead" form as shown in Fig.5, consisting of the following fields:

1. Contact Name: First Name, Last Name, Salutation.
2. Account Name:
3. Department:
4. Level:
5. Work Email:
6. Work- Phone:
7. Address: Street, Zip Code, City, Country (by default- The Netherlands).
8. Description:
9. SAVE button, CANCEL button, ADD CUSTOM FIELD Button.



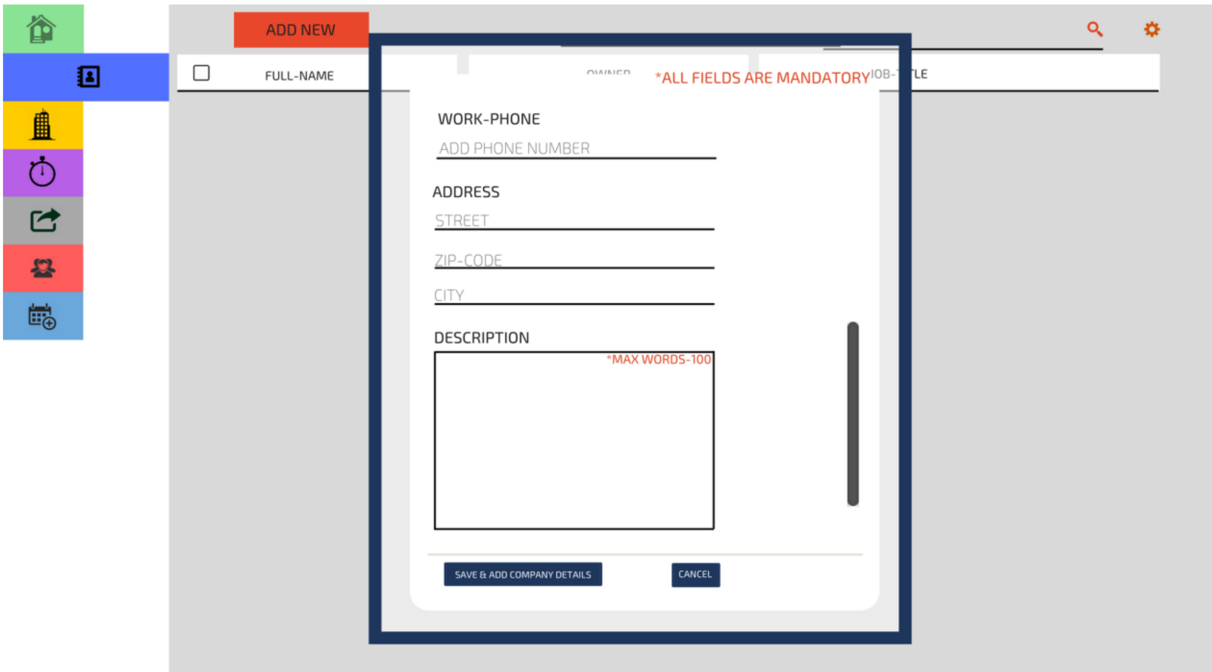


Fig.5

Search Bar.

Settings Button, which hosts the following buttons as shown in Fig.6:

1. The Customize Table button brings out a checkbox form that contains all the columns headers available alongside its corresponding checkbox if Active. It has a SAVE button and a CANCEL button at the end.
2. It also has a Filter Results button which brings out a side navigation menu.

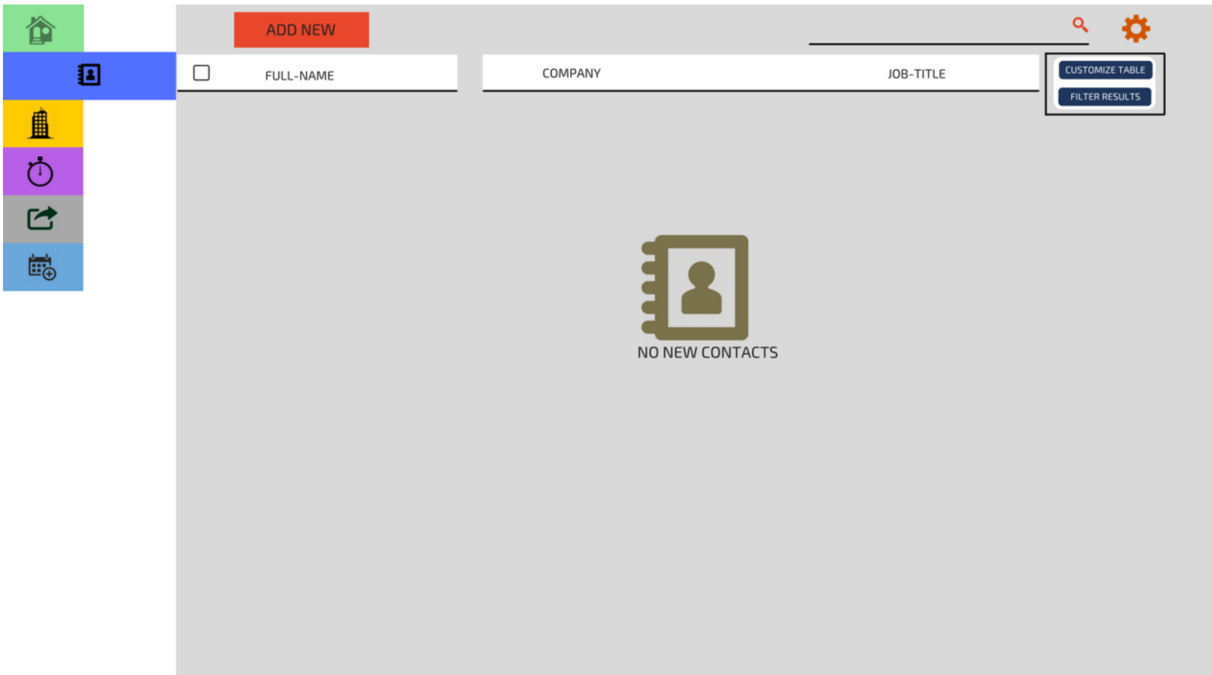


Fig.6a

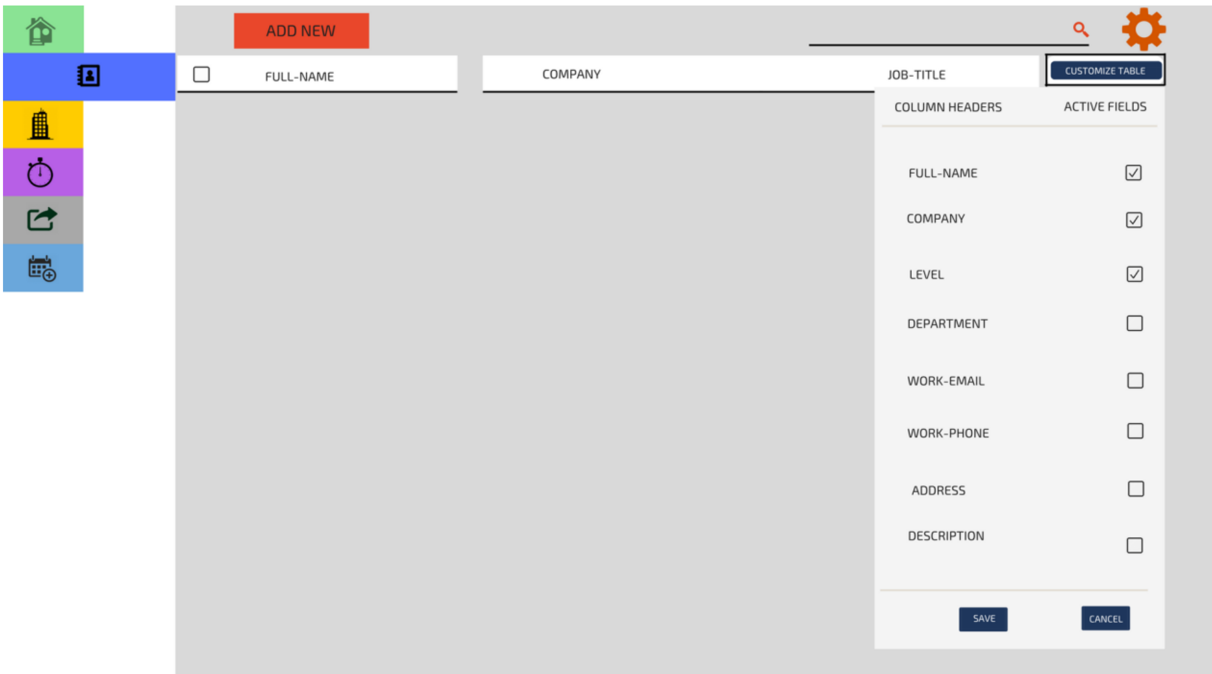


Fig.6b

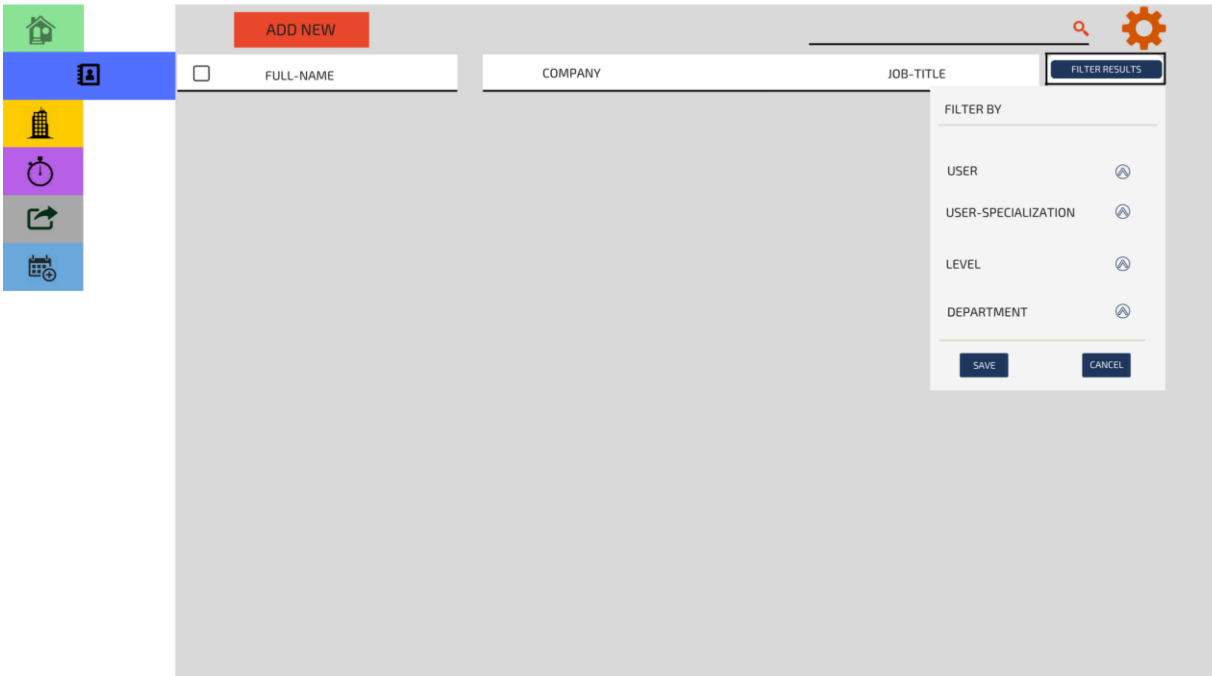


Fig.6c

After Checkbox State shows the following elements:

“ ” button.

“ ” button.

When all fields in the form are fulfilled then it appears as shown in Fig.7a.

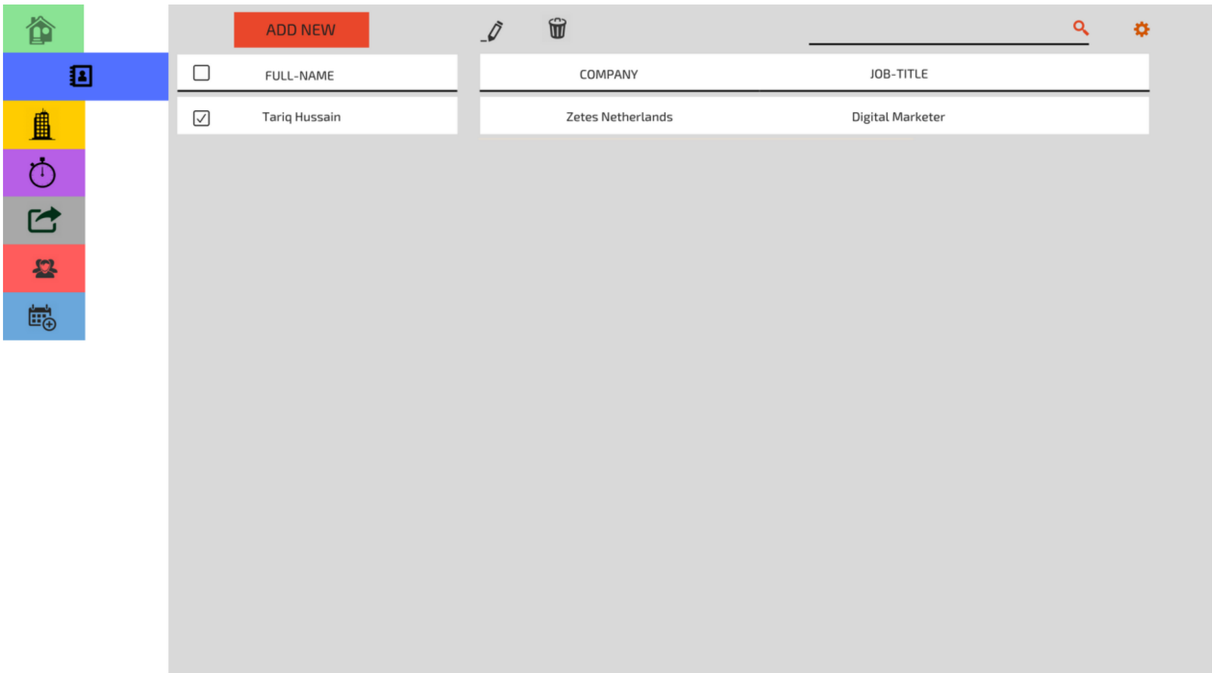


Fig.7a

Incomplete records are shown as in Fig.7b.

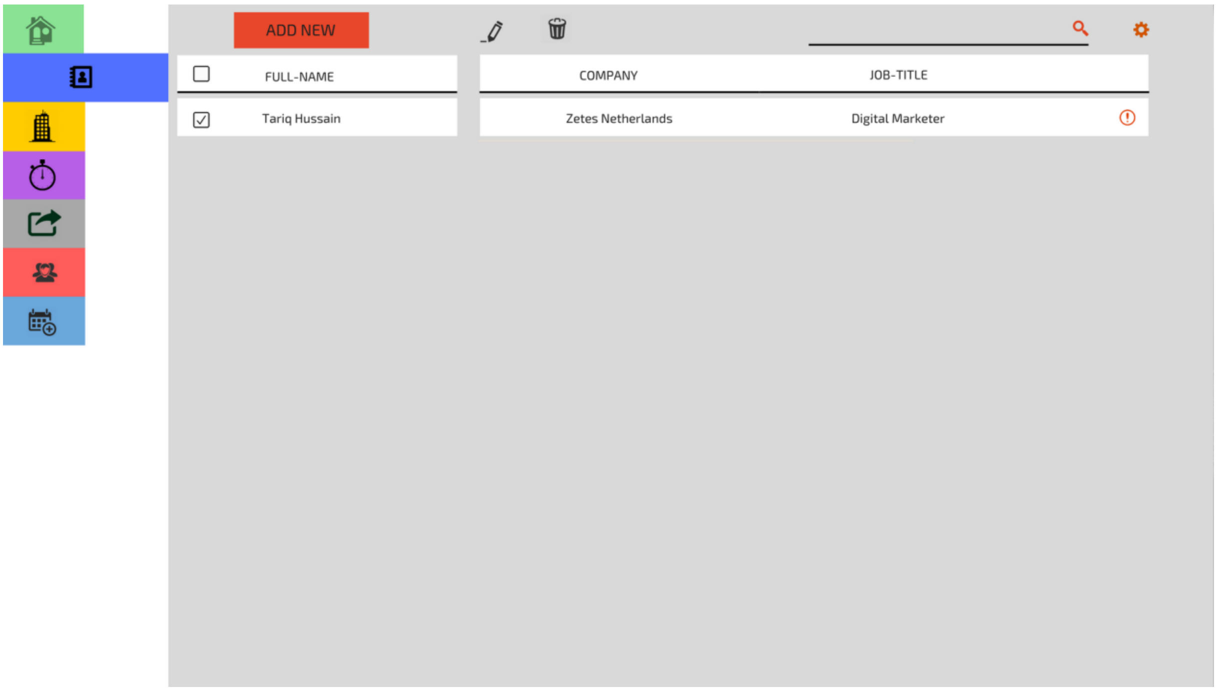


Fig.7b

- **ACCOUNTS-** This page shows a table which holds the following fields/column-headers which are fixed to view:
  Checkbox, Name, Industry, Associated Contact, Phone, see Fig.8.

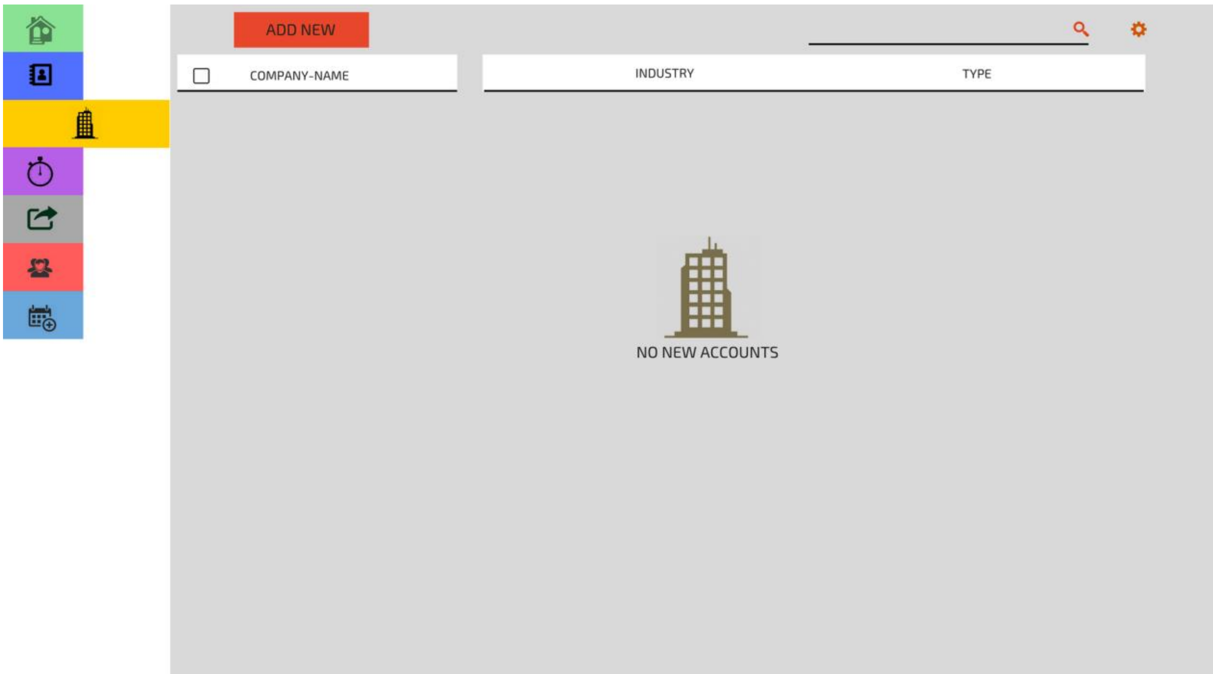
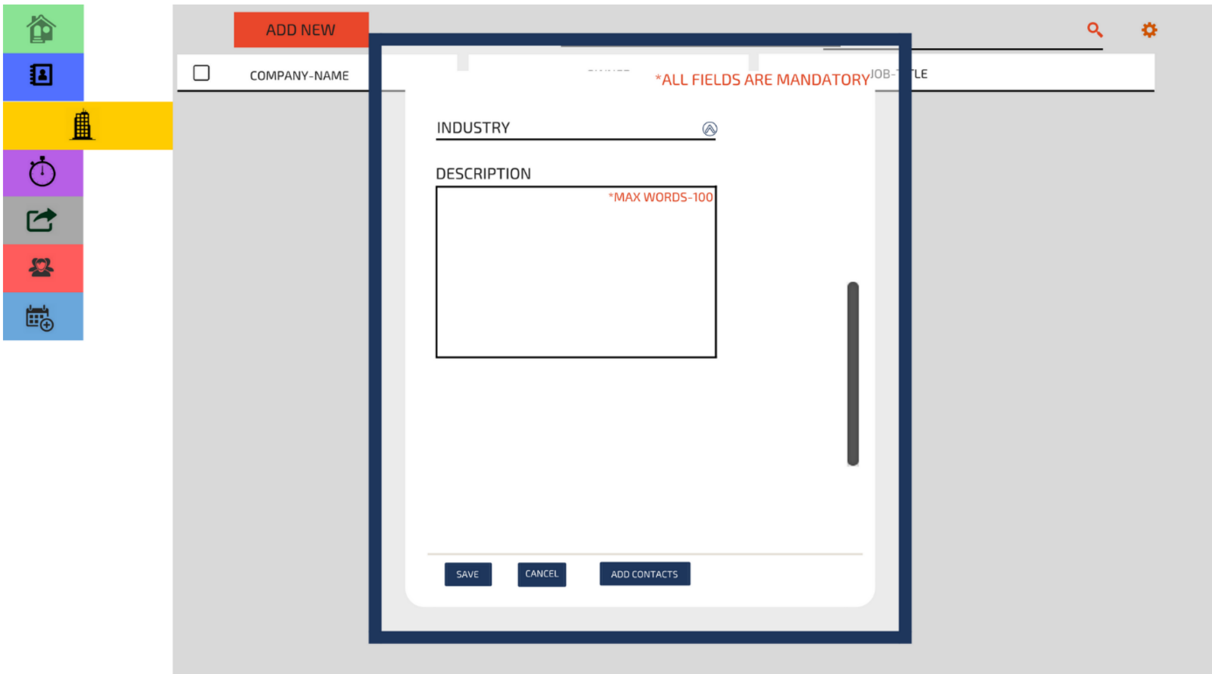


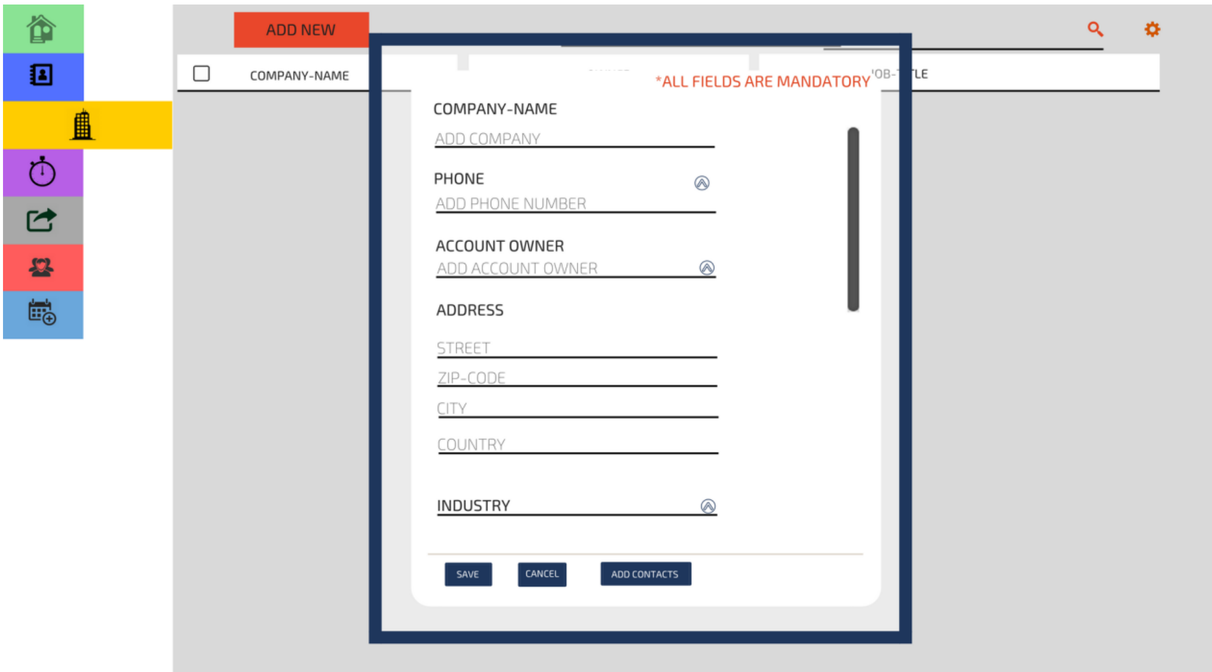
Fig.8

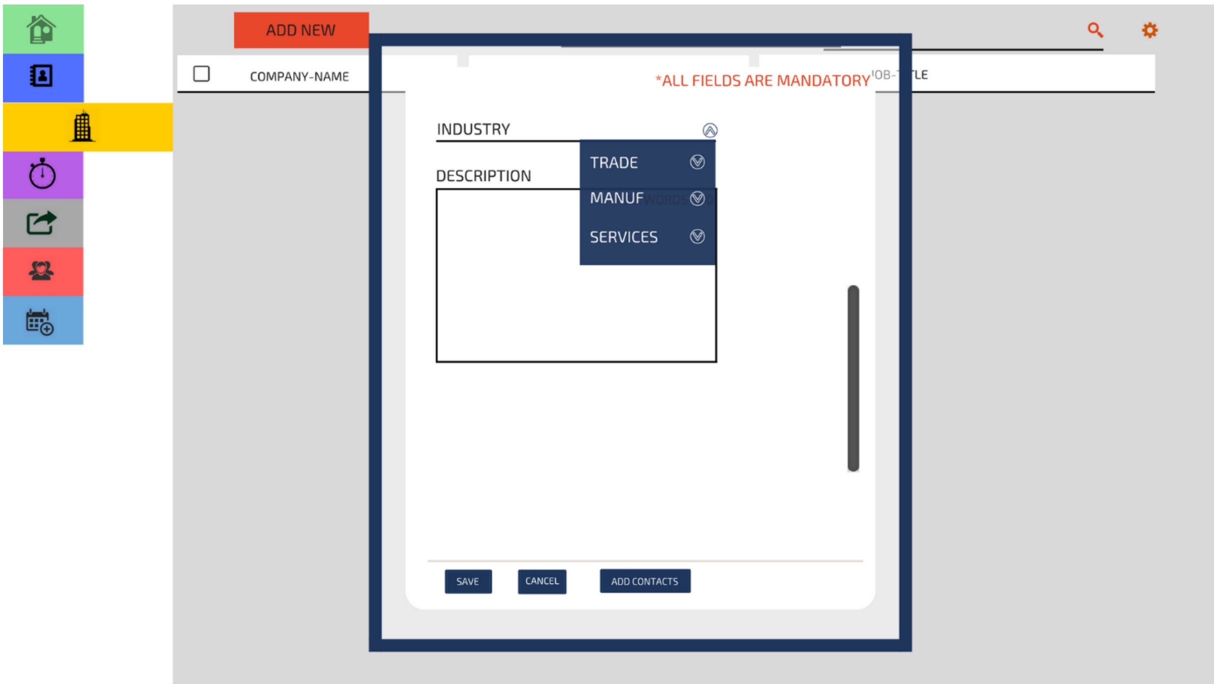
The horizontal menu bar above this table has two states, i.e. before and after records are check boxed. Before Checkbox State shows the following elements:

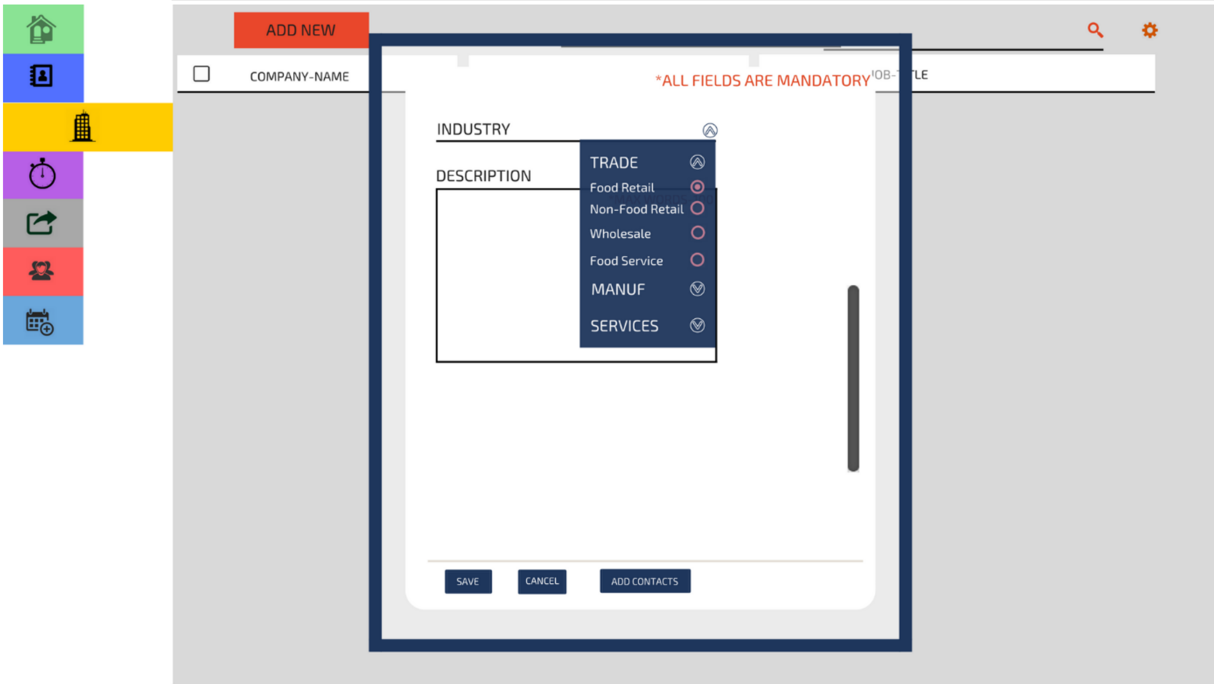
ADD NEW button brings out an "Add New Account" form as shown in Fig.9 consisting of the following mandatory-fill fields:

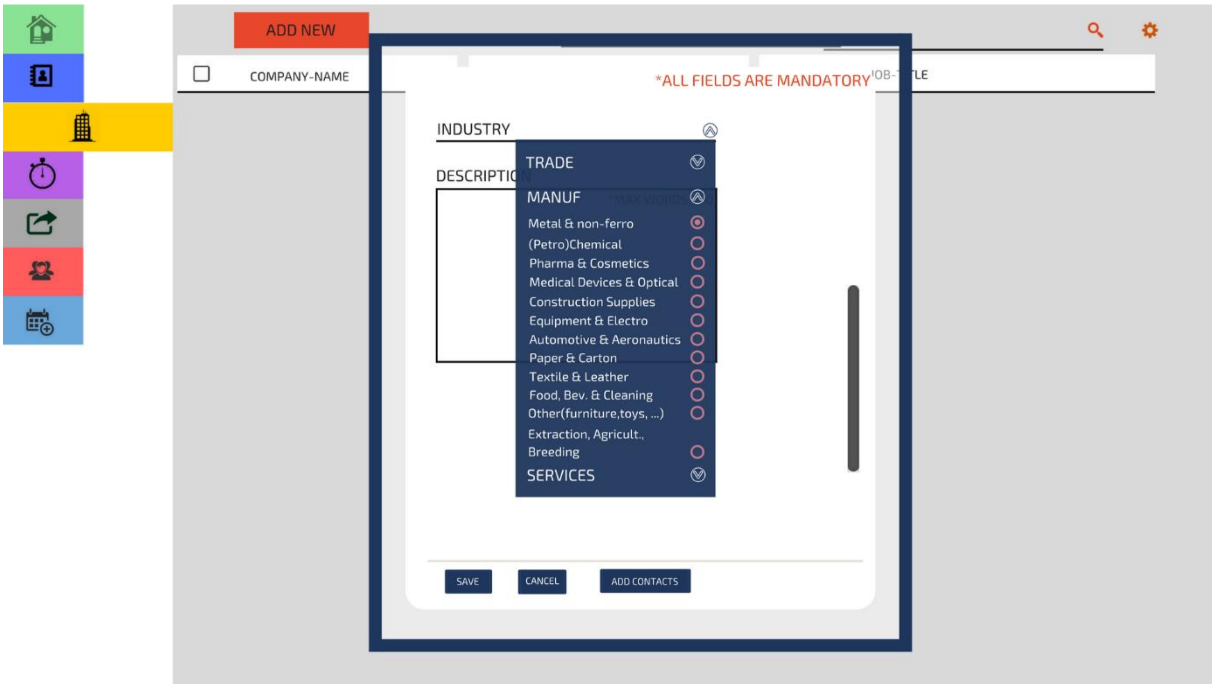
1. Company Name
2. Phone
3. Account Owner (Auto-fill with the name of the User)
4. Address: Street, Zip Code, City, Country (by default- The Netherlands)
5. Industry
6. Description
7. SAVE button and CANCEL button.






ADD NEW
COMPANY-NAME
*ALL FIELDS ARE MANDATORY
INDUSTRY
DESCRIPTION
TRADE
MANUF
SERVICES
SAVE
CANCEL
ADD CONTACTS


ADD NEW
COMPANY-NAME
*ALL FIELDS ARE MANDATORY
INDUSTRY
DESCRIPTION
TRADE
Food Retail
Non-Food Retail
Wholesale
Food Service
MANUF
SERVICES
SAVE
CANCEL
ADD CONTACTS



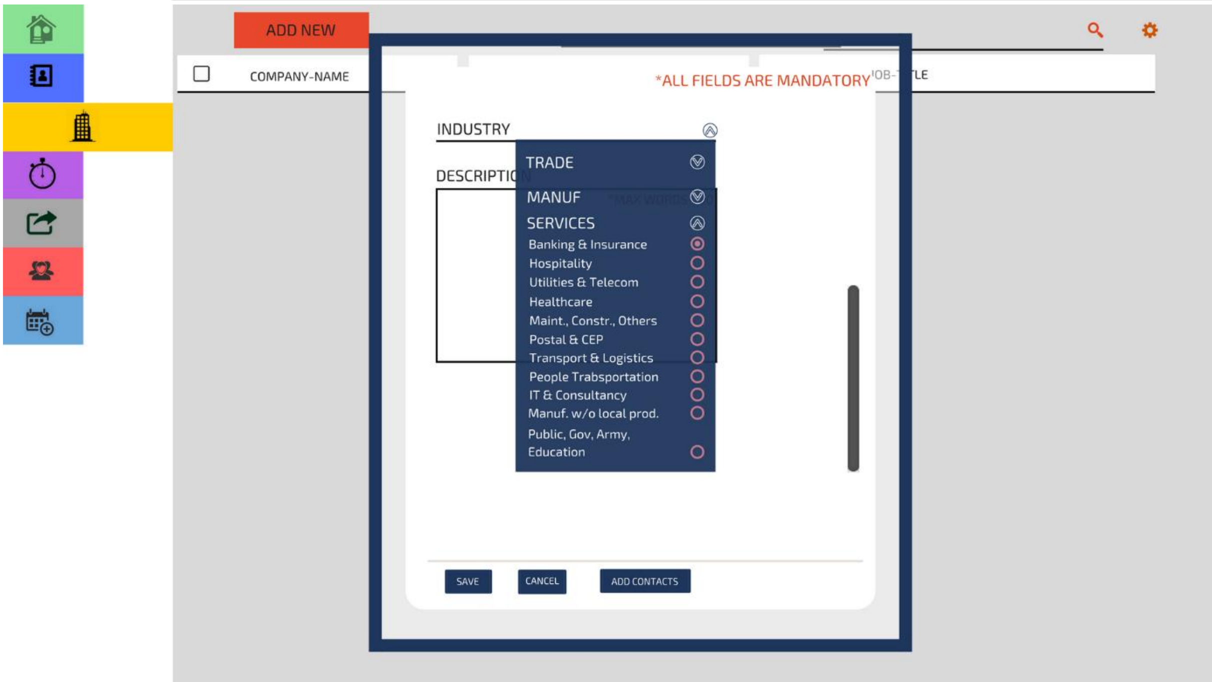


Fig.9

Search Bar

Settings Button which hosts the following selections:

1. The Customize Table button brings out a checkbox form that contains all the columns headers available alongside its corresponding checkbox. It has a SAVE button and a CANCEL button at the end.
2. The Add Contact Type button.

After Checkbox State shows the following elements:

"🗑" button.

"✎" button.

If the entry has been filled or Saved, then it appears as in Fig.10.

- **SESSIONS**- This page shows a table with the following fixed column headers: Checkbox, Session name, Start-Date, End-Date. Other column headers such as Leads, Contacts, Accounts, Status (in discrete levels- Running, Done) can be viewed through the 'Customize Table' in Settings, as shown in Fig.11.

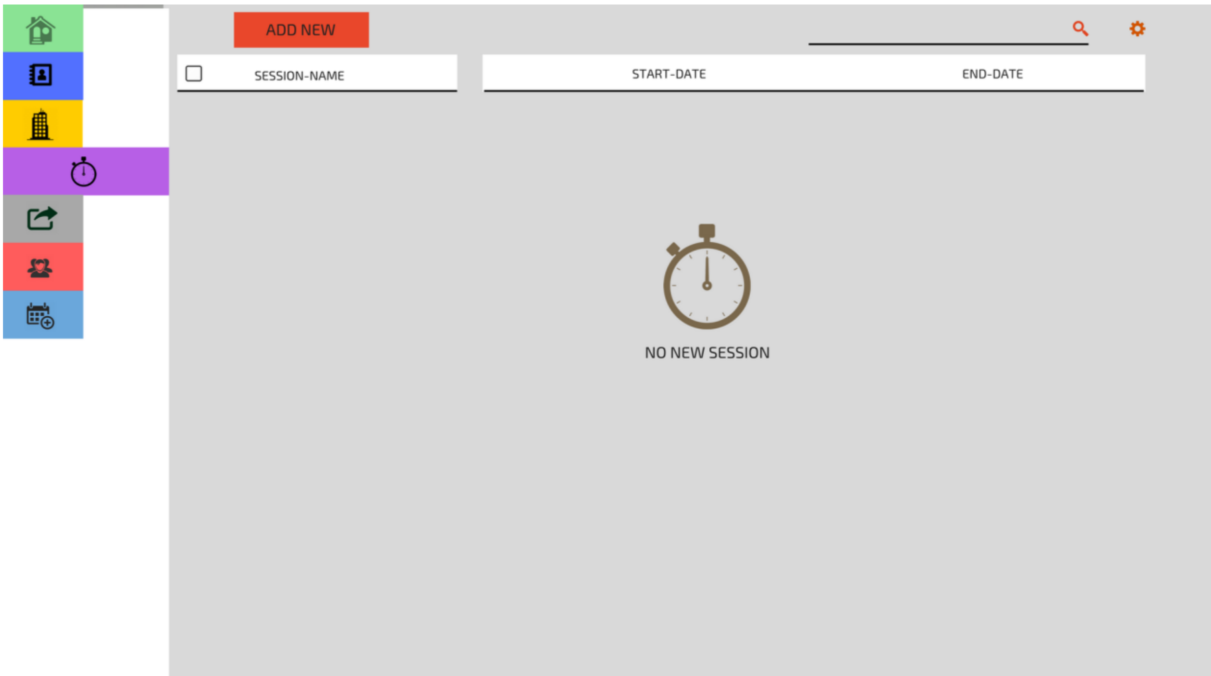


Fig.11

ADD NEW button brings out a "Add New Account" form as shown in Fig.12 consisting of the following mandatory-fill fields:

1. Session Name
2. Start Date
3. End Date
4. Duration in Day(s)
5. Add User Checklist
6. Associated Salesforce Campaign.
7. SAVE button and CANCEL button.

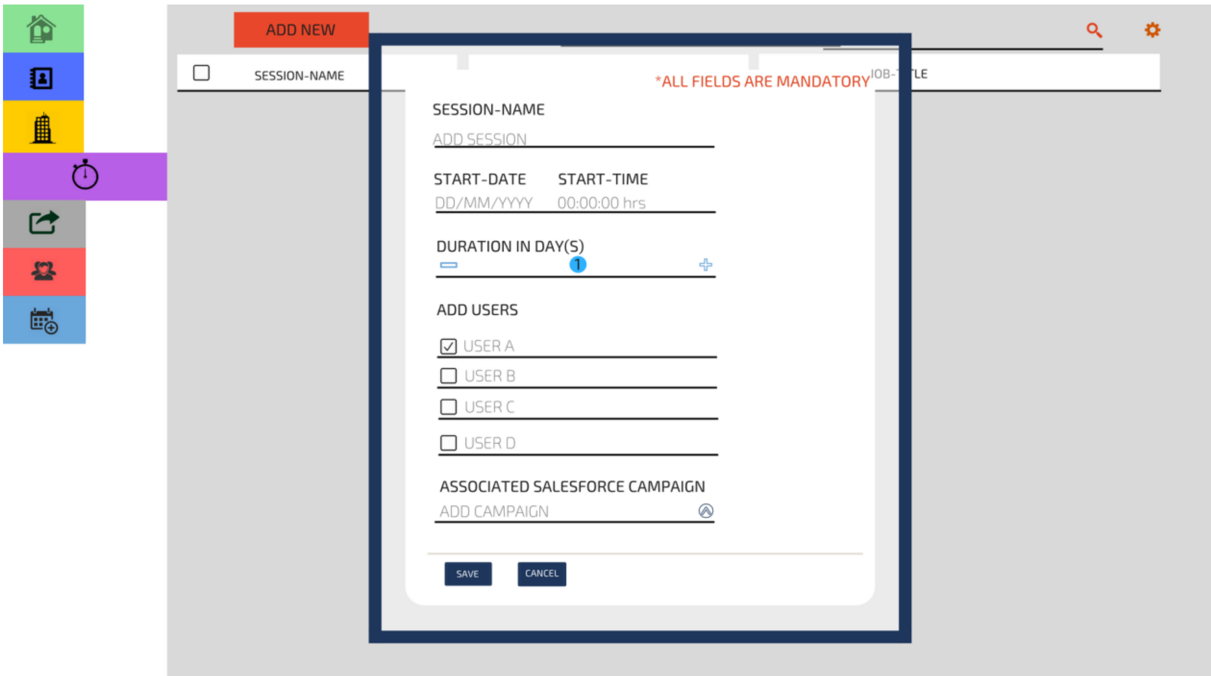


Fig.12

The horizontal menu bar above this table shows the following elements after records are check boxed, which are as follows:

" " button.

" " button.

After a Session is created, then it appears as shown in Fig.13a.

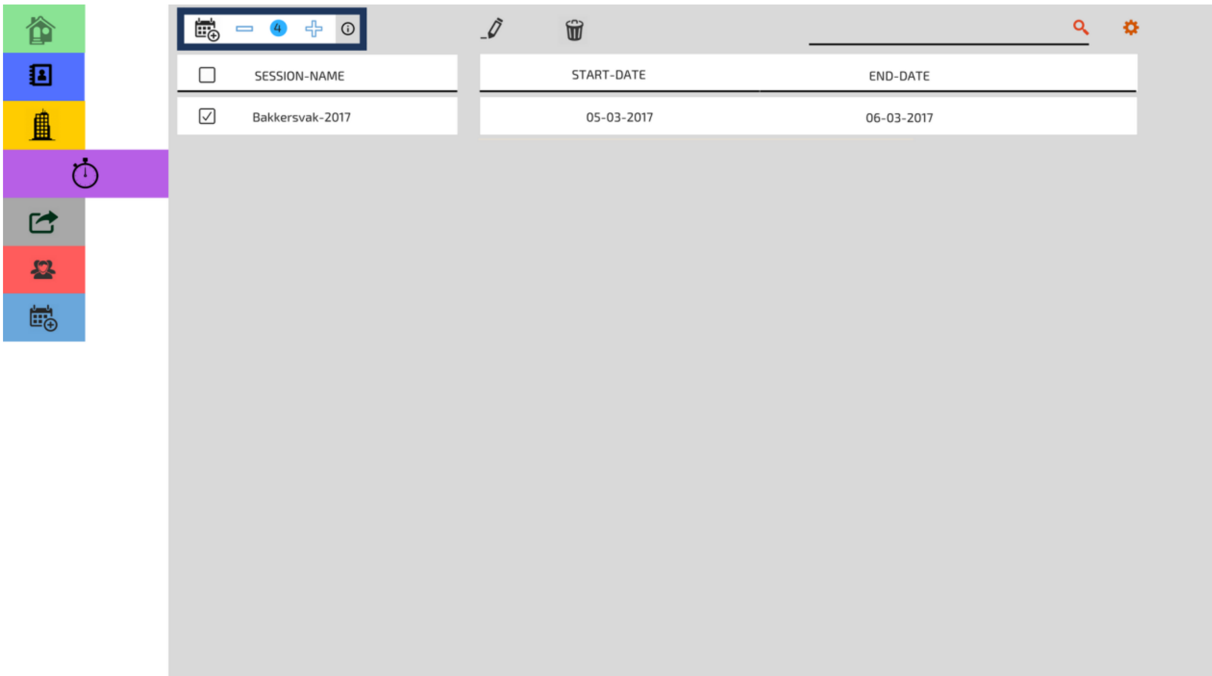


Fig.13

The User can request additional days to remain in a by Selecting a number above whole number above zero and clicking the info "ⓘ" button as pointed out by the red arrow in Fig.13b.

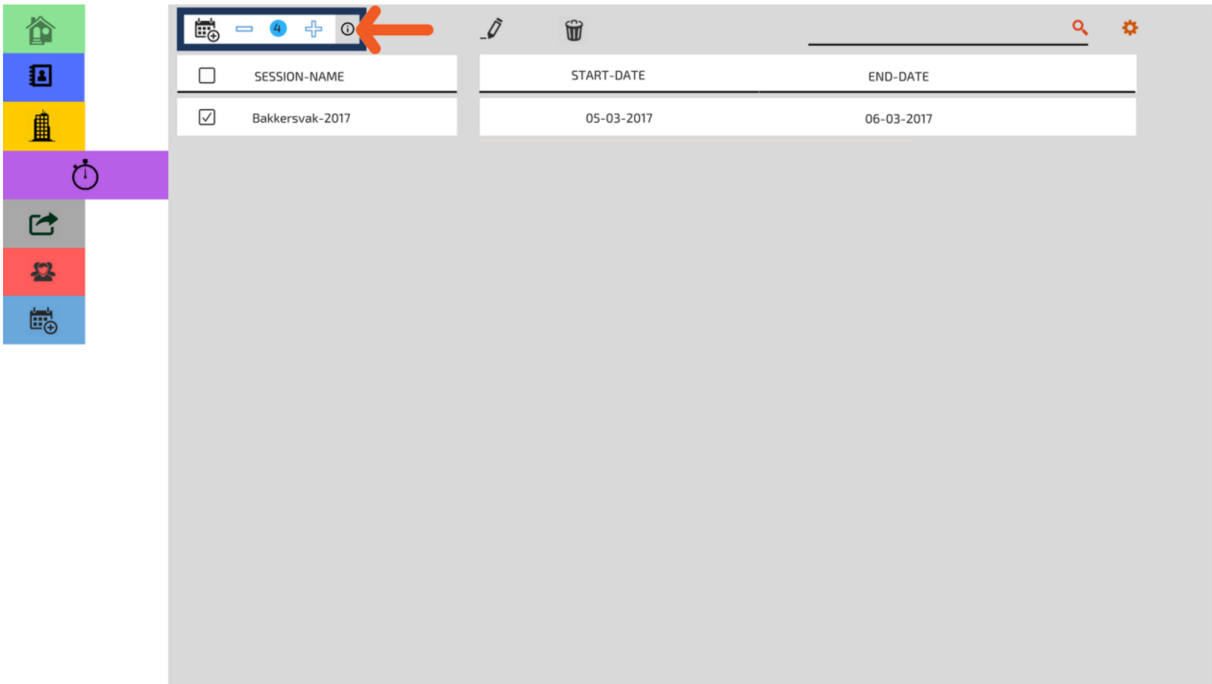


Fig 13b

This brings out a pop-up window with options "YES" or "NO", as shown in Fig.14.

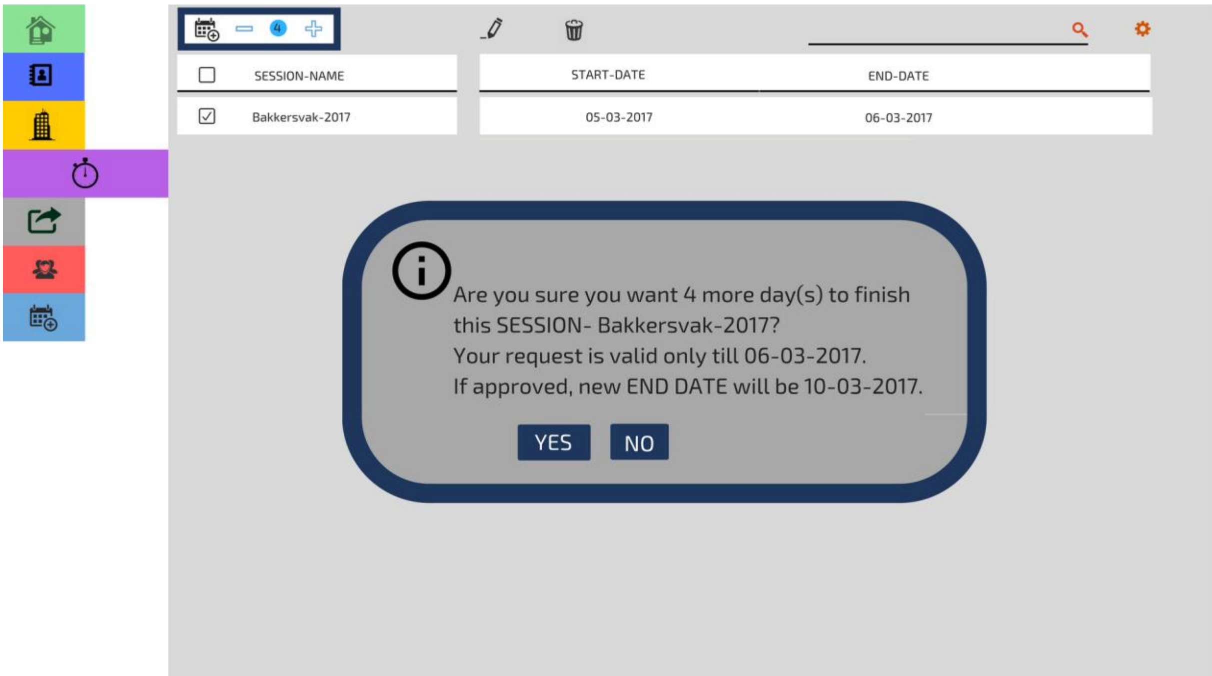


Fig.14

- **EXPORTS: -** This page shows a table which holds the following fields/column-headers which are fixed to view:

Checkbox, Export Name, Number of Accounts, Number of Contacts, Users, see Fig.15.

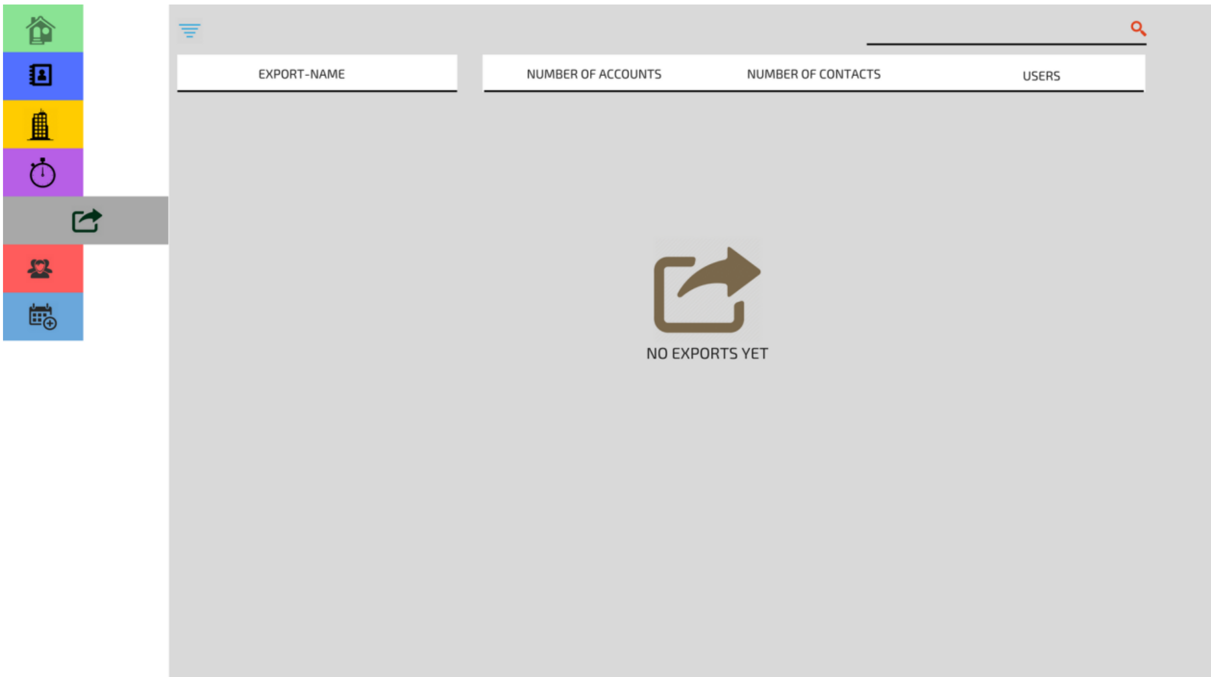


Fig.15

A successful export appears as shown in Fig.16.

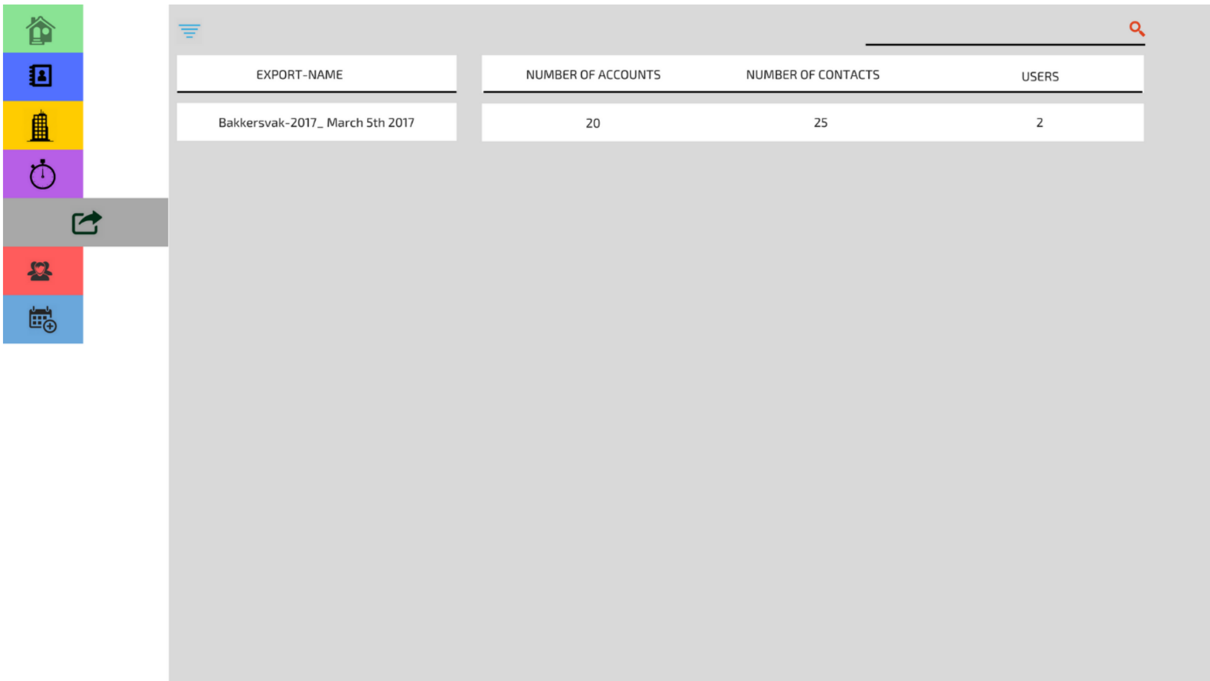


Fig.16

The horizontal menu bar above this table shows the Filter icon-

Clicking this icon will bring out the following menu as shown in Fig.17.

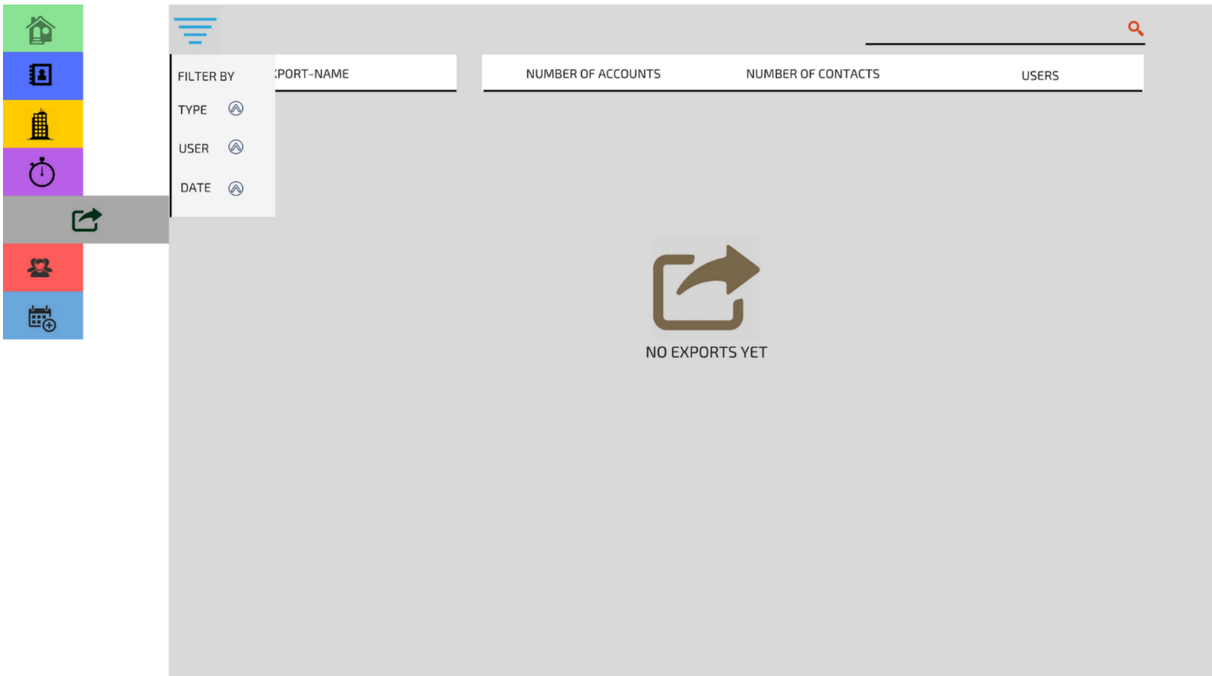


Fig.17a

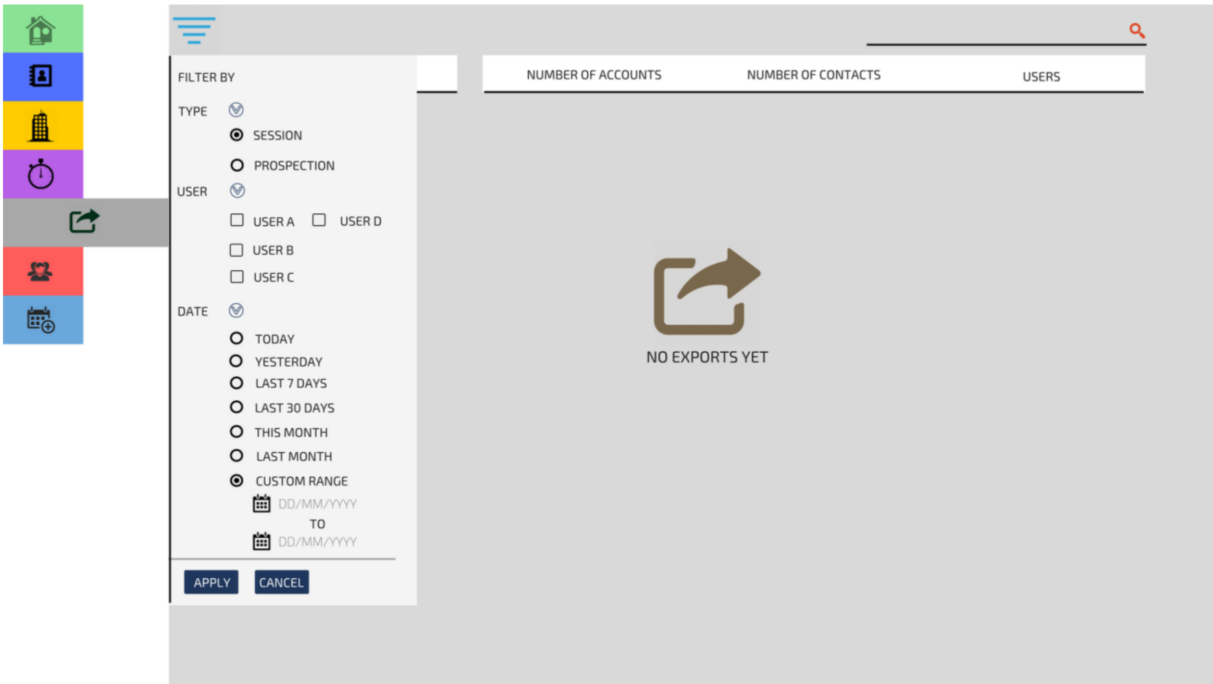


Fig.17b

- **USERS-**
  This page shows a table which holds the following fields/column-headers which are fixed to view: Checkbox, User-Name, Specialization, Phone, Email, see Fig.18.

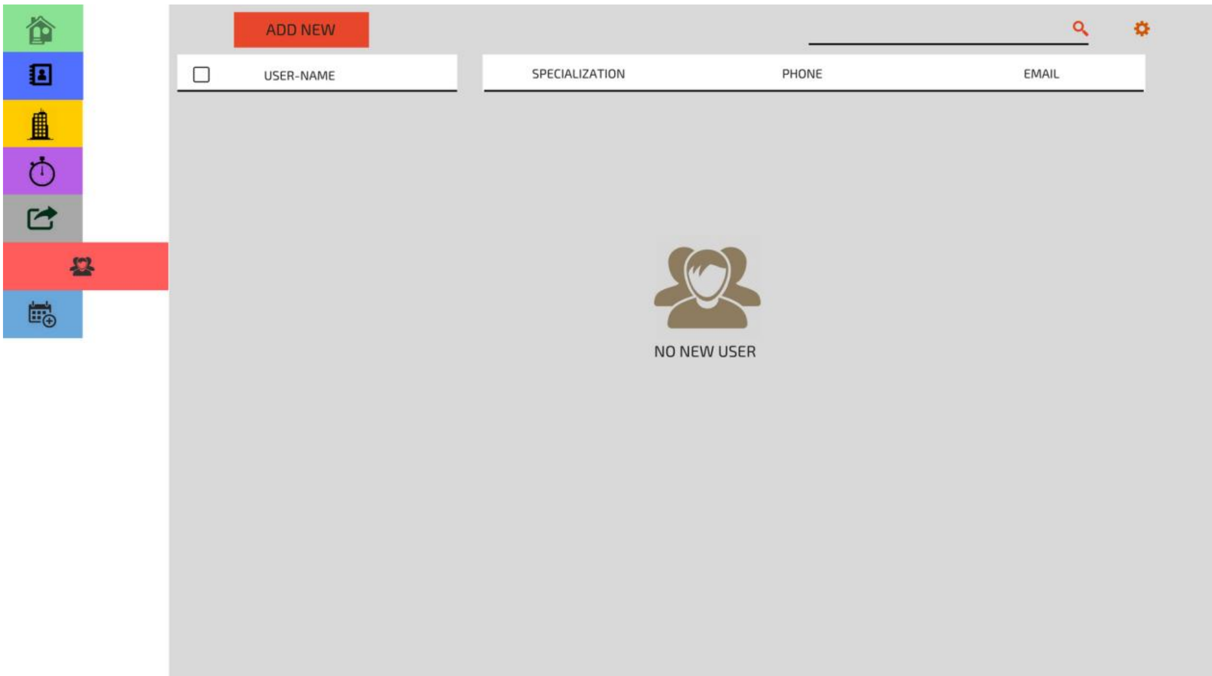


Fig.18

ADD NEW button brings out a "Add New User" form as shown in Fig.19 consisting of the following mandatory-fill fields:

1. First-Name
2. Last-Name
3. Specialization
4. Salesforce Username/Email
5. Phone
6. Country
7. ADD USER and CANCEL buttons.

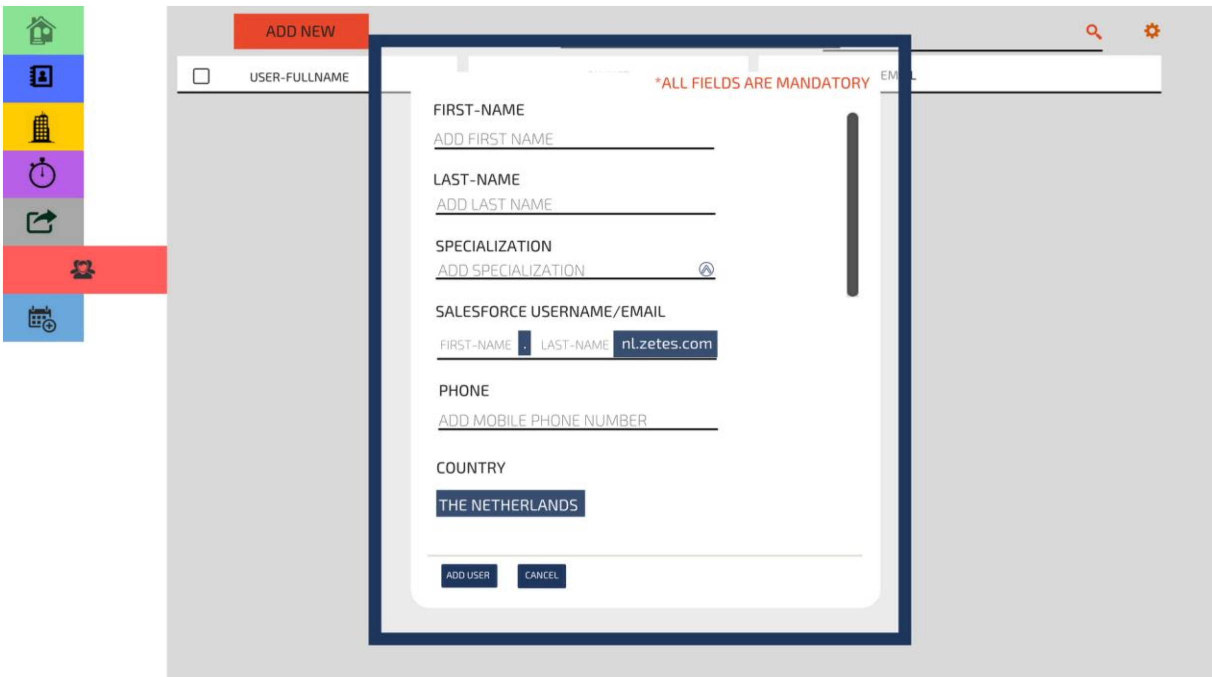


Fig.19

The SPECIALIZATION drop down brings the following options as shown in Fig.20:

1. Print & Apply
2. Printing & Medea
3. Internal Sales
4. General Sales

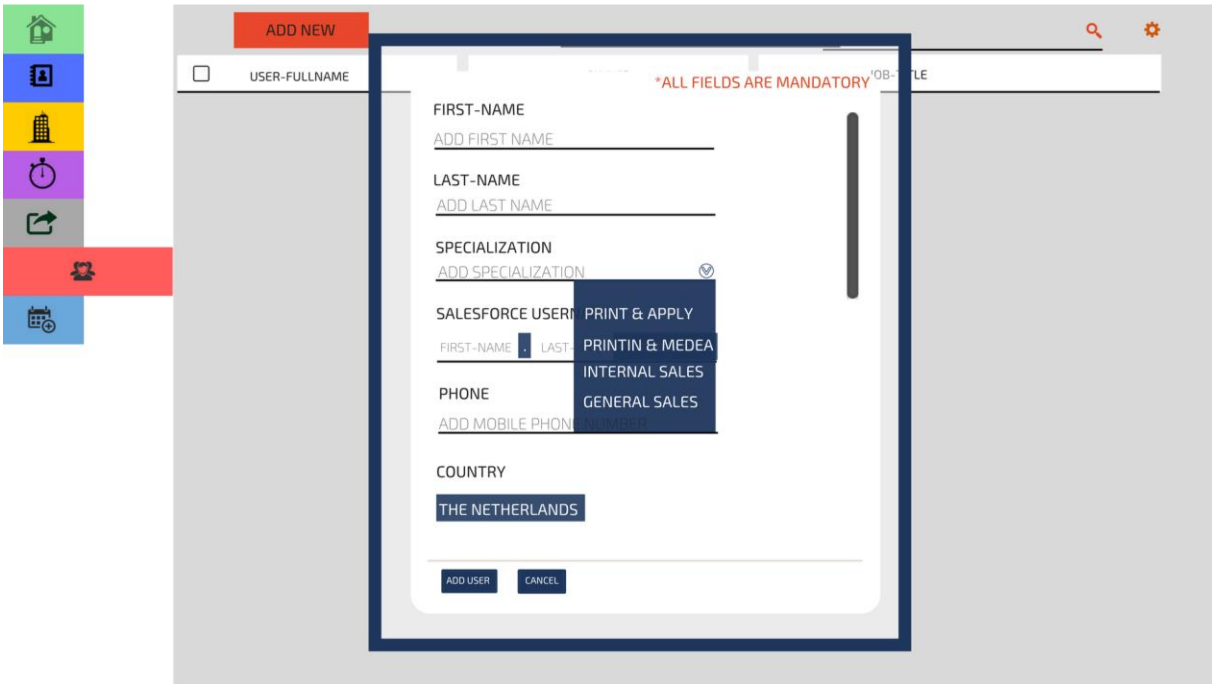


Fig.20

After the Administrator has created a USER, it is displayed as shown in Fig.21 below.

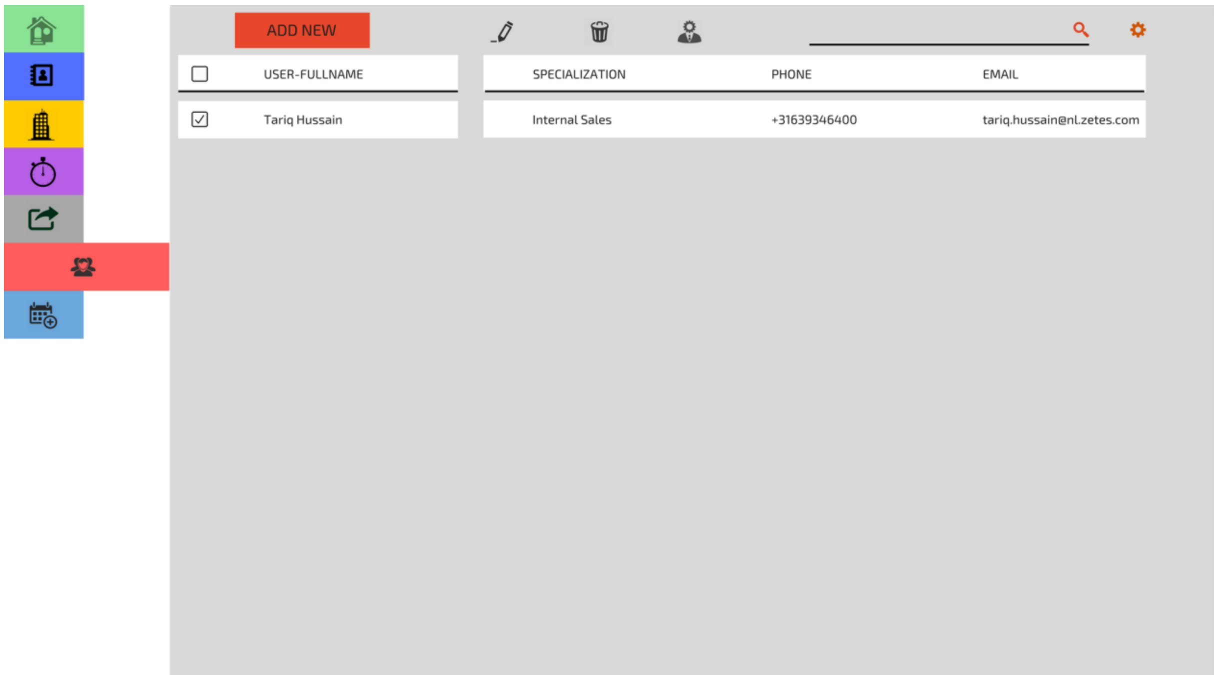


Fig.21

The horizontal menu bar above this table shows the following elements after records are check boxed, which are as follows:

“ ” button.

“ ” button.

“ ” button.

Clicking the button brings the following window as shown below in Fig.22.

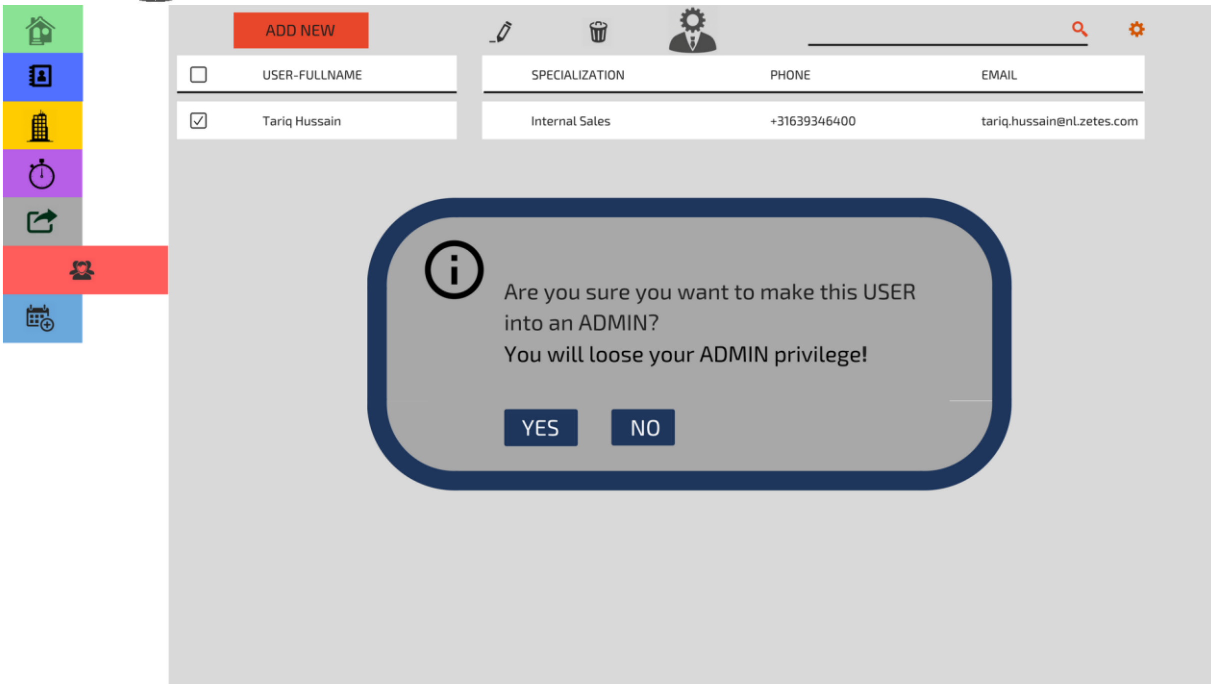


Fig.22

- **OVERTIME:** This page shows a table which holds the following fields/column-headers which are fixed to view:

Checkbox, User-Name, Session Name, Overtime in Days, Original Deadline, Extension Deadline as shown in Fig.23 below.

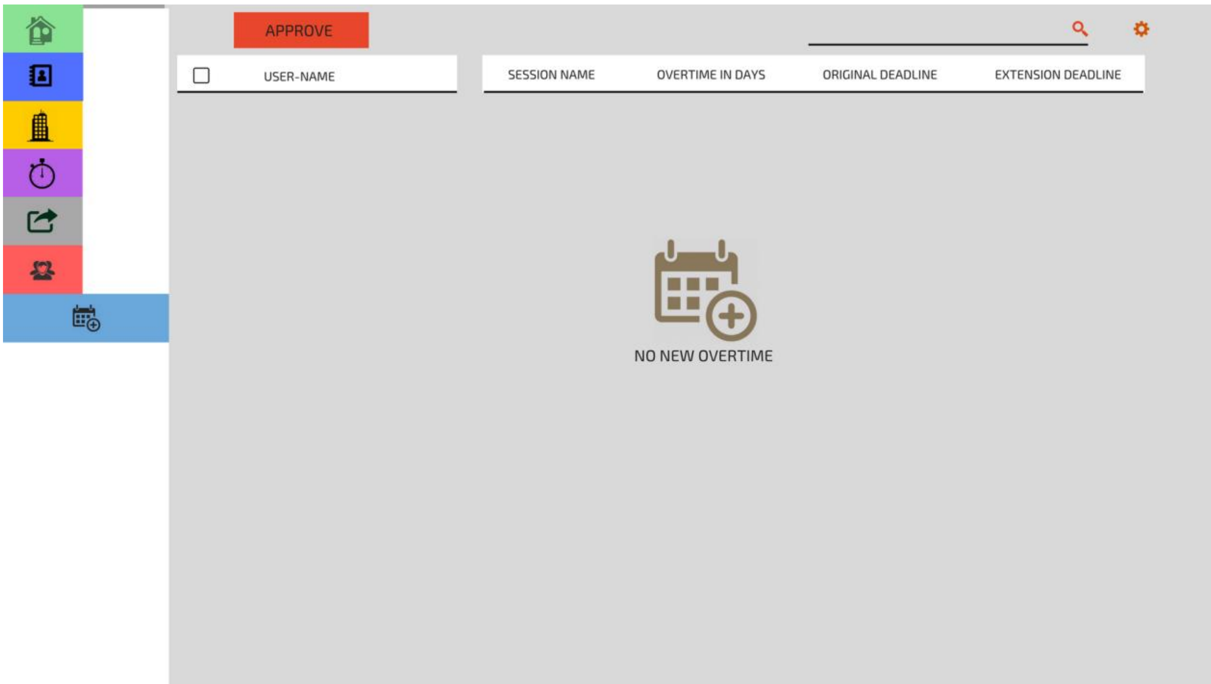


Fig.23

When a User makes a request for OVERTIME then the ADMIN may APPROVE the request by using the Checkbox as shown in Fig.24 below.

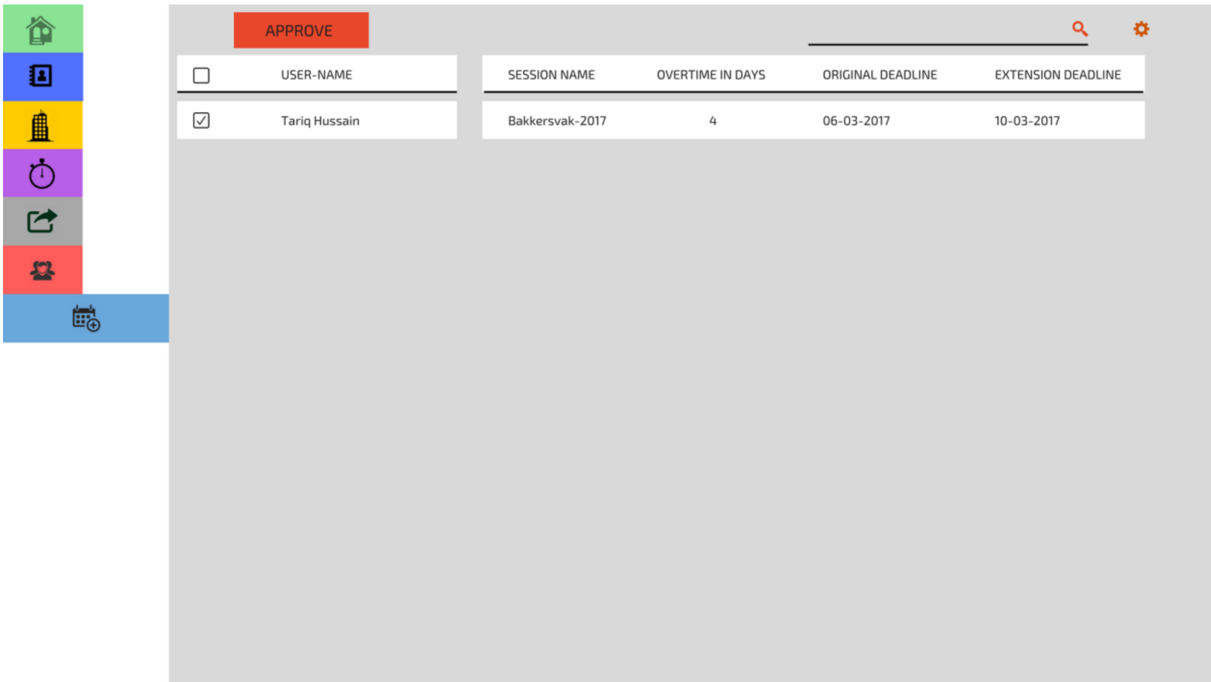
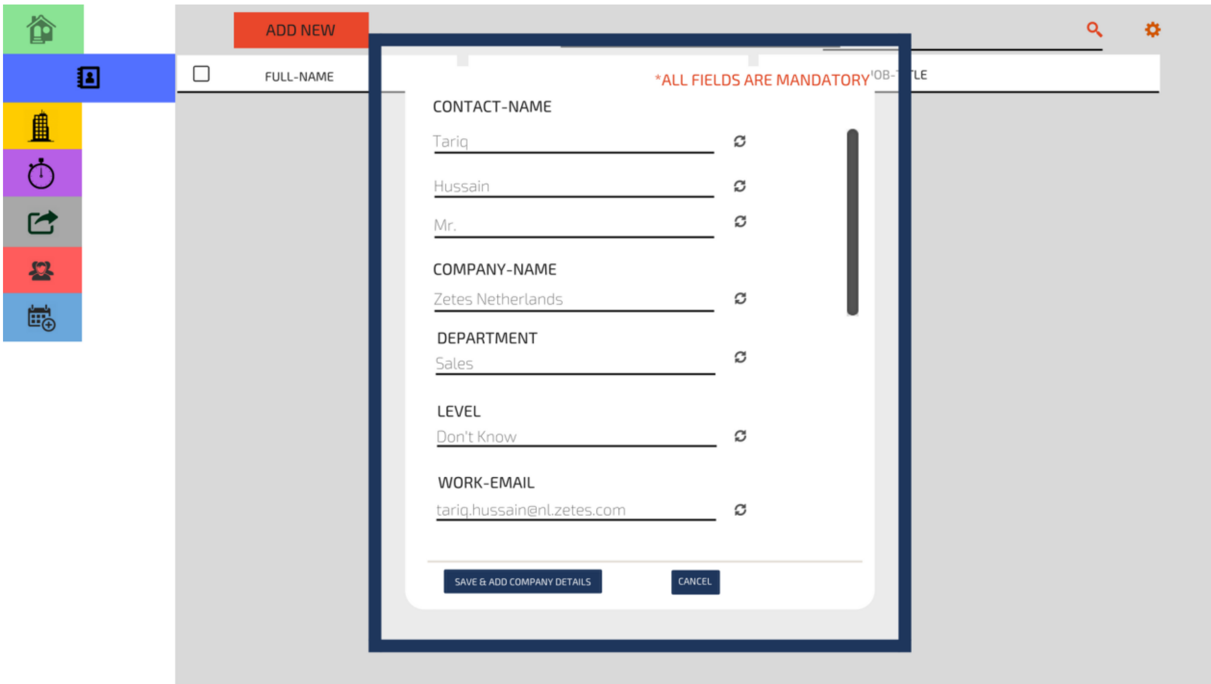


Fig.24

- **FILLING A FORM:**
  The User fills a Contact form in the way shown below in Fig.25.



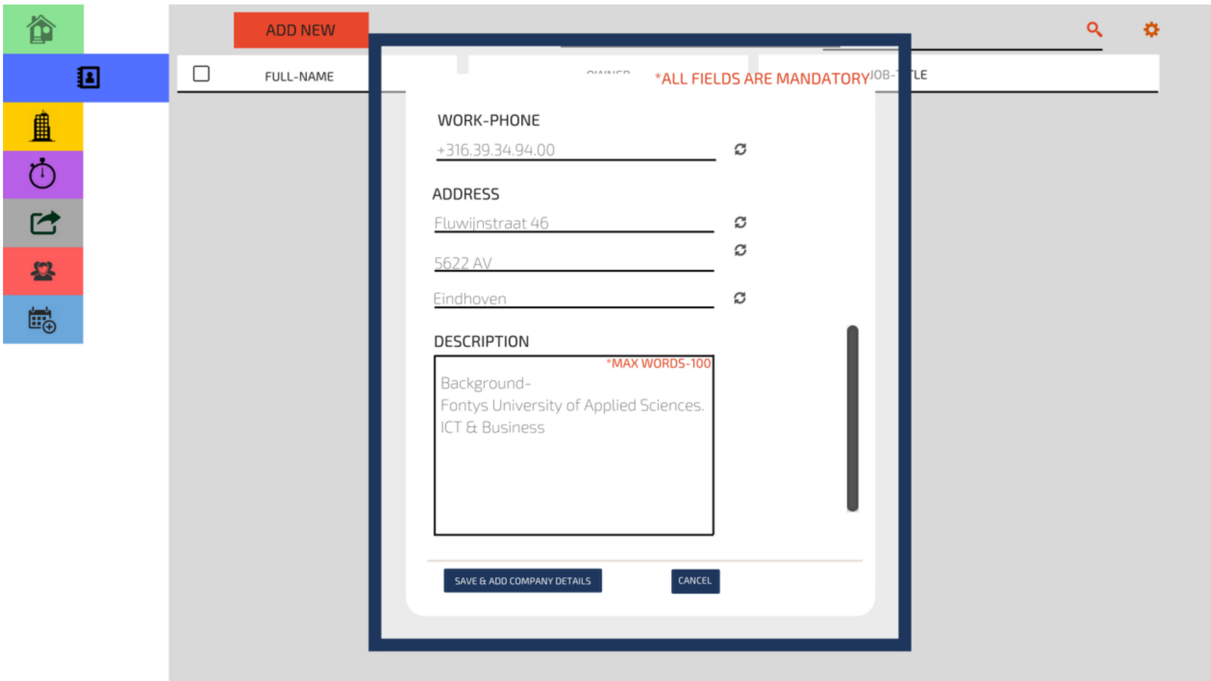


Fig.25

An existing contact in Salesforce cannot be placed as shown in Fig.26 below.

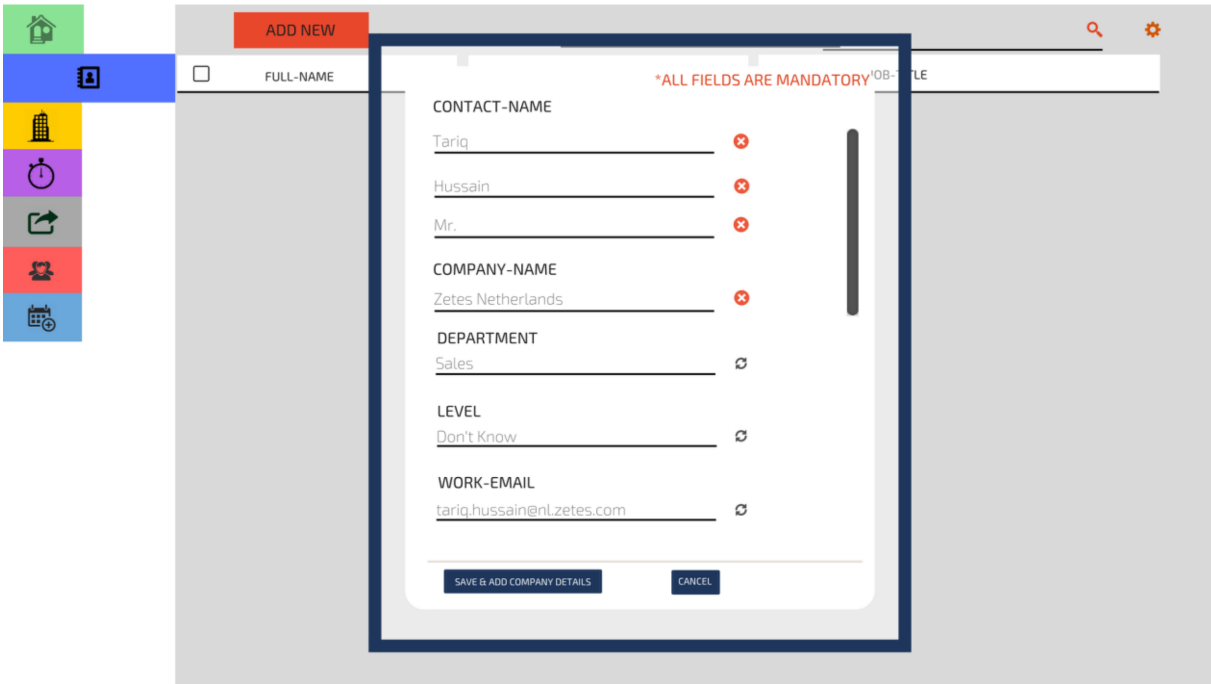


Fig.26

The User makes a choice to proceed in the directions provided as shown in Fig.27 below.

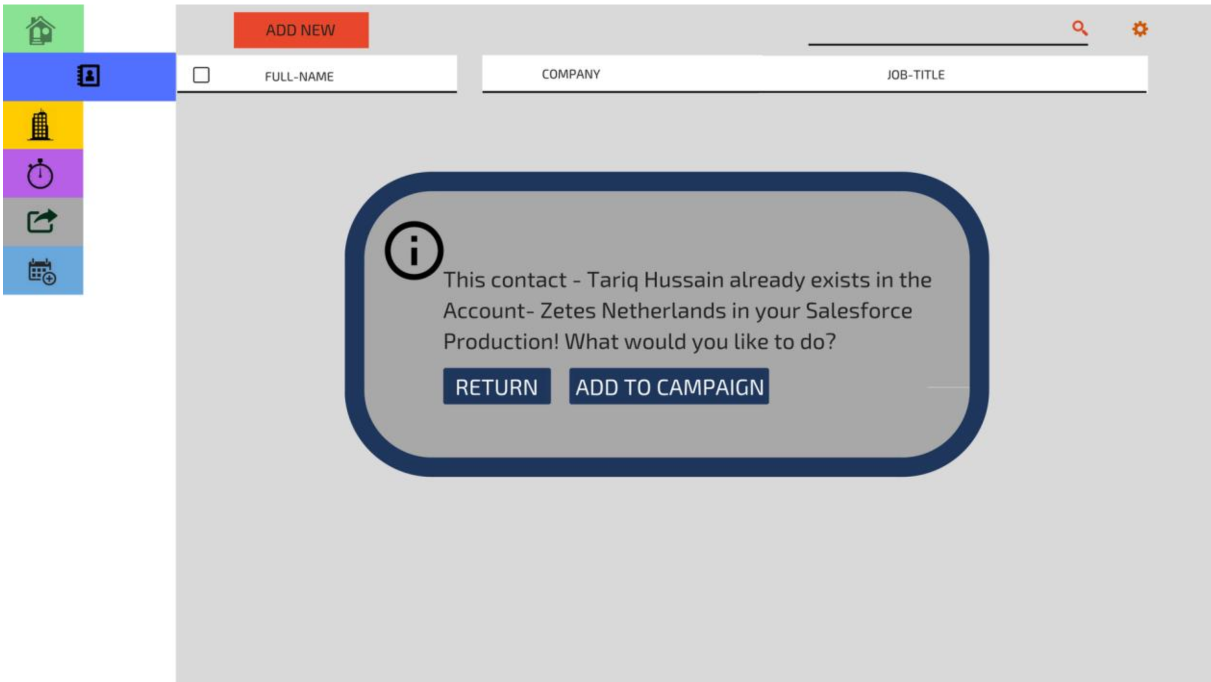
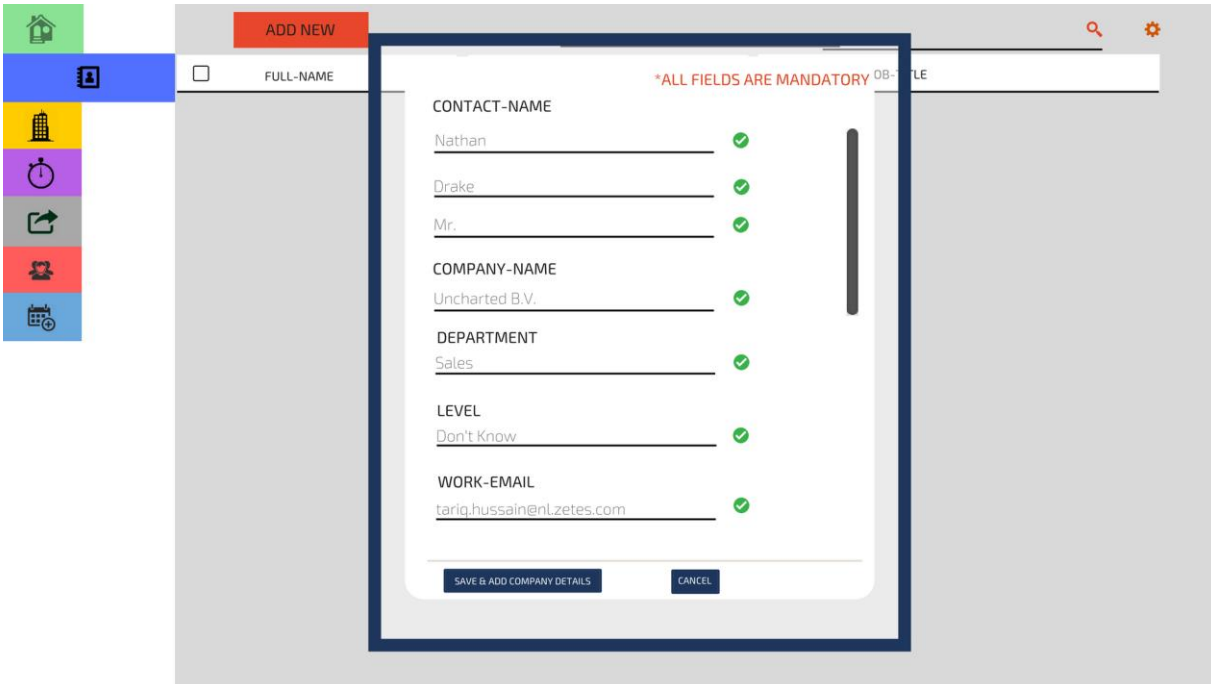


Fig.27

A new Contact will appear in the Contact submission form as shown below in Fig.28 below.



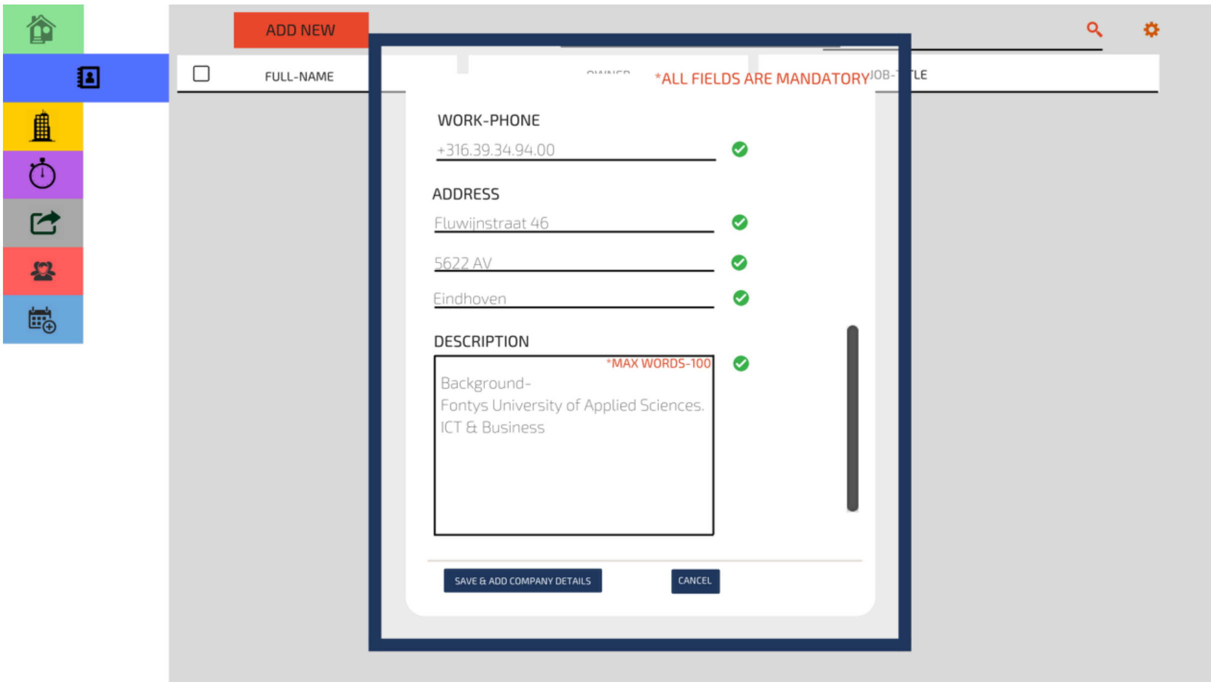


Fig.28

A new Account will appear in the Account submission form as shown below in Fig.29.

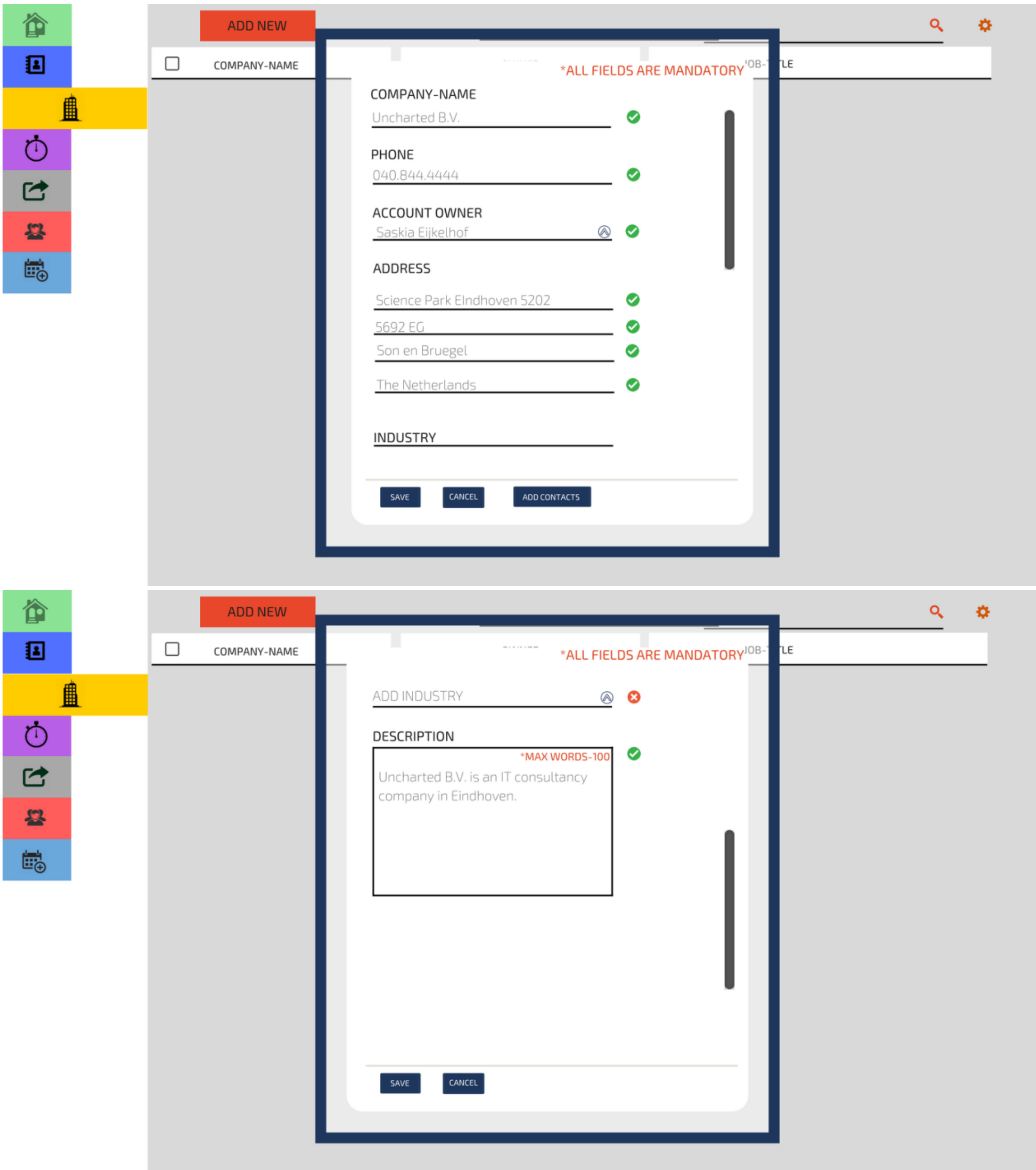


Fig.29

The NCRM allows the User of type “System Admin” the following controls over User of type “Data Export & Entry”

3.2 FUNCTIONAL REQUIREMENTS

In this section, all the fundamental actions performed by the software system are provided.

Functional Requirement 1

TITLE: Admin Registration Page and User and System Admin database.

DESCRIPTION: At the beginning there will be an Admin Registration form in the login page. Here using the Zetes email for a Username, a password and another criterion i.e. Country Division an Admin can be created. To support this, there will be a database with an Admin table and a User table.

DEPENDENCY: The email for Username should also be the same Username for their Salesforce account.

Functional Requirement 2

TITLE: Only one Admin for their Country Division/ Business Unit.

DESCRIPTION: In order to prevent someone from another country to become an Admin only one Admin per country division needs to be enforced.

DEPENDENCY: The first to register as Admin needs to be a Salesforce user for Zetes with Marketing privileges.

Functional Requirement 3

TITLE: Selecting country division triggers the association to their respective country email domain.

DESCRIPTION: If someone from Zetes Netherlands seeks to be an admin then his/her email domain i.e. nl.zetes.com in this case, needs to be used in the Admin Registration form.

DEPENDENCY: He/she should have the same Salesforce username.

Functional Requirement 4

TITLE: An Admin from a Country Division/ Business Unit may only add Users from his/her Country Division.

DESCRIPTION: If an Admin from Zetes Netherlands creates a User then that User will also be from Zetes Netherlands.

DEPENDENCY: The User also is a Salesforce user.

Functional Requirement 5

TITLE: An Admin can only switch to a User after declaring another User as an Admin for their respective Country Division.

DESCRIPTION: An Admin from Zetes Netherlands can become a User only after declaring another User from Zetes Netherlands as an Admin.

DEPENDENCY: The Admin already added someone as a User.

Functional Requirement 6

TITLE: Admin Login

DESCRIPTION: The Admin can login using their Salesforce Username and Password as shown in Fig.1.

DEPENDENCY: The Admin has registered using the Admin Registration page.

Functional Requirement 7

TITLE: User Login

DESCRIPTION: The User can login using their Salesforce Username as shown in Fig.2.

DEPENDENCY: The User is created by the Admin before login.

Functional Requirement 8

TITLE: User and Admin login exclusivity.

DESCRIPTION: An Admin cannot login as a User and a User cannot login as an Admin.

DEPENDENCY: Functional Requirement 2

Functional Requirement 9

TITLE: Users may enter their Job-Title, Department in the Home page.

DESCRIPTION: As shown in Fig.3 a User may enter personal details like Job-Title they have and or the Department they belong to.

DEPENDENCY: None.

Functional Requirement 10

TITLE: The phone number of a User is shown in Home page as entered by the Admin.

DESCRIPTION: As shown in Fig.3 a User's phone number is already filled. This is the number which was entered by the Admin while creating the User – Tariq Hussain i.e. +31.6.39.34.64.00

DEPENDENCY: The phone number field is not interactive.

Functional Requirement 11

TITLE: The Home page displays the Prospection name and Session details to a User and the Admin.

DESCRIPTION: As shown in Fig.3a a User can see the Prospection and Session he/she is associated. Also the User can see other Users associated with that Session in a dropdown menu and the Start-End dates.

DEPENDENCY: The Admin has already added that the User to a Session.

Functional Requirement 12

TITLE: Every User is associated to the Prospection by default.

DESCRIPTION: If a User is not associated to a Session then that User's activity is under the Prospection.

DEPENDENCY: The Admin creates the Prospection prior to creating any Sessions.

Functional Requirement 13

TITLE: The Admin fixes the Prospection.

DESCRIPTION: The Admin can fix the Prospection by using the dropdown menu listing all the Campaigns that are currently running in Salesforce, as shown in Fig.3b.

DEPENDENCY:

- The campaign used for the Prospection is available for use.
- The campaign is not associated to a Session.

Functional Requirement 14

TITLE: User/Admin can stop and restart a Session before it ends.

DESCRIPTION: As shown in Fig.3a the User can choose to STOP a running Session and switch to Prospection mode before the Session reaches its end-date. The User can also RESTART that Session.

DEPENDENCY: The Session has not reached its end-date.

Functional Requirement 15

TITLE: User/Admin adds Contacts.

DESCRIPTION: A User or the Admin adds a Contact using the form as shown in Fig.5.

DEPENDENCY: A User or the Admin needs to click the ADD NEW button also shown in Fig.5.

Functional Requirement 16

TITLE: Check for duplicates in Salesforce.

DESCRIPTION: The form will check for duplicate contacts in Salesforce as shown in Fig.25.

DEPENDENCY: A User or the Admin fills the minimum of the Contact First Name, Last Name and Company Name.

Functional Requirement 17

TITLE: Duplicates are shown with a red cross.

DESCRIPTION: When the form detects duplicate entries then a red cross is placed at its side as shown in Fig.26.

DEPENDENCY: Functional Requirement 15

Functional Requirement 18

TITLE: Choice of Fill and Return or add to running campaign or Session.

DESCRIPTION: After the duplicate is detected then the User/Admin is presented with a choice as shown in Fig.27. The Choices are “FILL & RETURN” and “ADD TO CAMPAIGN”

DEPENDENCY: Functional Requirement 16

Functional Requirement 19

TITLE: The “FILL & RETURN” choice.

DESCRIPTION: If the User/Admin selects the option “FILL & RETURN” after filling out the Contact form, then the form searches that contact in Salesforce to check if any corresponding fields are missing any data. If so then they are filled out in Salesforce using with these entries. After that the User is returned to the HOME page.

DEPENDENCY: Functional Requirement 17

Functional Requirement 20

TITLE: The “ADD TO CAMPAIGN” choice.

DESCRIPTION: If the USER/ADMIN selects the option “ADD TO CAMPAIGN” after filling out the form, then the form searches that contact in Salesforce to check if any corresponding fields are missing any data. If so then they are filled out in Salesforce using these entries. After that the Contact is added to the Salesforce campaign associated to the running Session or the Prospection depending on what the User/Admin is using at that moment.

DEPENDENCY: Functional Requirement 17

Functional Requirement 21

TITLE: All fields are mandatory in the Contact form.

DESCRIPTION: Only forms with all fields fulfilled qualify for submission to Salesforce.

DEPENDENCY: The application checks if all fields are fulfilled.

Functional Requirement 22

TITLE: The FIRST-NAME and LAST-NAME fields in the Contact form.

DESCRIPTION: The FIRST-NAME and LAST-NAME fields in the Contact form are mapped to the “First Name” and “Last Name” objects in the “Contacts” object in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 23

TITLE: The SALUTATION drop-down menu in the Contact form.

DESCRIPTION: In the SALUTATION drop-down menu, the User/Admin can select one of the following items.

- Female.
- Male.

DEPENDENCY:

- SALUTATION is mapped to the Gender object in the Contacts object in Salesforce.
- Miss/Mrs. is mapped to the Female object in the Gender object in Salesforce.
- Mr. is mapped to the Male object in the Gender object in Salesforce.
- The application integrates with Salesforce.

Functional Requirement 24

TITLE: The COMPANY-NAME field in the Contact form.

DESCRIPTION: The COMPANY-NAME field in the Contact form is mapped to the "Account Name" object in the "Contacts" object in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 25

TITLE: The DEPARTMENT drop-down menu in the Contact form.

DESCRIPTION: In the DEPARTMENT drop-down menu, the User/Admin can select one of the following items:

- eCommerce / Marketing
- Engineering
- Finance
- General
- InStore
- IT
- Logistics
- Maintenance
- Packaging
- Production
- Purchasing
- Quality
- Sales
- Supply Chain
- Transport / Distribution
- Warehouse

DEPENDENCY: DEPARTMENT is mapped to the "Department" object in the "Contacts" object in Salesforce.

Functional Requirement 26

TITLE: The LEVEL drop-down menu in the Contact form

DESCRIPTION: In the LEVEL drop-down menu, the User/Admin can select one of the following items:

- Director
- Manager
- Collaborator

DEPENDENCY:

- LEVEL is mapped to the "Level" object in the "Contacts" object in Salesforce.
- The application integrates with Salesforce.

Functional Requirement 27

TITLE: The WORK-EMAIL field in the Contact form.

DESCRIPTION: The WORK-EMAIL field in the Contact form is mapped to the "Email" object in the "Contacts" in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional requirement 28

TITLE: The WORK-PHONE field in the Contact form

DESCRIPTION: The WORK-PHONE field in the Contact form is mapped to either “Phone” object or the “Mobile” object in the “Contacts” object in Salesforce.

DEPENDENCY:

- The application integrates with Salesforce.
- If the first two digits contains the numbers 0 and 6 then it is mapped to the “Mobile” object, otherwise it is mapped to the “Phone” object.

Functional Requirement 29

TITLE: The STREET, ZIP-CODE and CITY fields in the Contact form.

DESCRIPTION: The STREET, ZIP-CODE and CITY field in the Contact form are mapped to the “Mailing Street”,” Mailing Zip/Postal Code” and “Mailing City” objects in the “Contacts” object respectively.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 30

TITLE: The DESCRIPTION field in the Contact form.

DESCRIPTION: The DESCRIPTION field in the Contact form is mapped to the “Description” object in the “Contacts” object in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 31

TITLE: The SAVE & ADD COMPANY DETAILS button and the CANCEL button in the form.

DESCRIPTION: The User can click the SAVE & ADD COMPANY DETAILS button and the CANCEL button after filling out the form as shown in Fig.28

DEPENDENCY:

- The form finds no duplicate for ACCOUNT-NAME in Salesforce.
- The form may or may not be filled for the CANCEL button to function.

Functional Requirement 32

TITLE: The SAVE & ADD COMPANY DETAILS button functionality when the form is complete.

DESCRIPTION: When the User clicks the SAVE & ADD COMPANY DETAILS button after completing the form, then the ACCOUNT form as shown in Fig.9 follows and all the fields in the Contact form a record which appears as shown in Fig.7a.

DEPENDENCY: Functional Requirement 31

Functional Requirement 33

TITLE: The SAVE & ADD COMPANY DETAILS button functionality when the form is incomplete.

DESCRIPTION: When the User clicks the SAVE & ADD COMPANY DETAILS button without completing the form then the ACCOUNT form as shown in Fig.9 follows and all the fields in the Contact form a record which appears as shown in Fig.7b.

DEPENDENCY: Functional Requirement 31

Functional Requirement 34

TITLE: The CANCEL button functionality.

DESCRIPTION: When the User clicks the CANCEL button then the form clears the fields and cache. The Contact form appears as shown in Fig.5

DEPENDENCY: At least one of the fields are filled.

Functional Requirement 35

TITLE: The SETTINGS button in the CONTACTS page.

DESCRIPTION: By clicking the SETTINGS button in the CONTACTS page makes it bigger and brings out a menu with two options as shown in Fig.6a.

- Customize Table
- Filter Results

DEPENDENCY: None.

Functional Requirement 36

TITLE: The CUSTOMIZE TABLE button.

DESCRIPTION: By clicking the CUSTOMIZE TABLE button the User is given the list of Column Headers available for selection to view by check-box as shown in Fig.6b.

DEPENDENCY: The User clicks the SETTINGS button.

Functional Requirement 37

TITLE: The FILTER RESULTS button & menu.

DESCRIPTION: By clicking the FILTER RESULTS button the User is given the possibility to filter the Contacts by User, User-Specialization, Level and Department as shown in Fig.6c. In the FILTER RESULTS menu there is also the SAVE and CANCEL buttons.

DEPENDENCY: The User clicks the SETTINGS button.

Functional Requirement 38

TITLE: The SAVE button in the FILTER RESULTS menu.

DESCRIPTION: By clicking the SAVE button the User asks the application to fix the filter criterion set by the User.

DEPENDENCY: The User sets the filter criterion before using the SAVE button.

Functional Requirement 39

TITLE: The CANCEL button in the FILTER RESULTS menu.

DESCRIPTION: By clicking the CANCEL button the User asks the application to reset the filter criterion that the User made.

DEPENDENCY: The User/Admin takes at least one of the four parameters in the filter criterion.

Functional Requirement 40

TITLE: Reset the existing filter criterion.

DESCRIPTION: If a User/Admin decides to change a filter criterion then the User can do so by using another set of parameter(s) and clicking the SAVE button or delete the current criterion by clicking the CANCEL button.

DEPENDENCY: The new criterion should be different from the old criterion.

Functional Requirement 41

TITLE: The User/Admin add an Account.

DESCRIPTION: The User/Admin can add an Account by using the form as shown in Fig.9.

DEPENDENCY: The User/Admin clicks the ADD NEW button as shown in Fig.9.

Functional Requirement 42

TITLE: The ACCOUNT-NAME field in Account form.

DESCRIPTION: The ACCOUNT-NAME field in the Account form is mapped to the “Account Name” object in the “Accounts” object in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 43

TITLE: The PHONE field in the Account form.

DESCRIPTION: The PHONE field in the Account form is mapped to the” Phone” object in the “Accounts” object in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 44

TITLE: The ACCOUNT OWNER drop-down menu in the Account form.

DESCRIPTION: In the ACCOUNT OWNER drop-down menu the User/Admin can select the User to which the account should belong to depending on the Session or Prospection.

DEPENDENCY:

- The Admin has created the User.

- The Admin has added some Users to a Session.

Functional Requirement 45

TITLE: The STREET, ZIP-CODE, CITY and COUNTRY fields in the Account form.

DESCRIPTION: The STREET, ZIP-CODE, CITY and COUNTRY fields are mapped to the "Billing Street", "Billing Zip/Postal Code", "Billing City", "Billing Country" objects in the "Accounts" object in Salesforce respectively.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 46

TITLE: The INDUSTRY drop-down menu in the Account form.

DESCRIPTION: In the INDUSTRY drop-down menu, the User/Admin can select the following items, as shown in Fig.9:

- TRADE drop down menu
- MANUF drop down menu
- SERVICES drop down menu

In the TRADE drop down menu, the User/Admin can select the following radio-button items:

  - Trade - Food Retail
  - Trade – Non-Food Retail
  - Trade – Wholesale
  - Trade – Food Service

In the MANUF drop down menu, the User/Admin can select the following radio-button items:

  - Manuf – Metal & non-ferro
  - Manuf – (Petro)Chemical
  - Manuf – Pharma & Cosmetics
  - Manuf – Medical Devices & Optical
  - Manuf – Construction Supplies
  - Manuf – Equipment & Electro
  - Manuf – Automotive & Aeronautics
  - Manuf – Paper & Carton
  - Manuf – Textile & Leather
  - Manuf – Food, Bev. & Cleaning
  - Manuf – Other (furniture, toys, …)
  - Extraction, Agricult., Breeding

In the SERVICES drop down menu, the User/Admin can select the following radio-button items:

  - Services – Banking & Insurance
  - Services – Hospitality
  - Services – Utilities & Telecom
  - Services – Healthcare
  - Services – Maint., Constr., Others
  - Services – Postal & CEP

- Services – Transport & Logistics
- Services – People Transportation
- Services – IT & Consultancy
- Services – Manuf. w/o local prod.
- Public, Gov, Army, Education

The items here are listed as in Salesforce and they should be mapped accordingly.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 47

TITLE: The DESCRIPTION field in the Account form.

DESCRIPTION: The DESCRIPTION field in the Account form is mapped to the “Description” object in the “Accounts” object in Salesforce.

DEPENDENCY: The application integrates with Salesforce.

Functional Requirement 48

TITLE: The SAVE and CANCEL buttons in the Account form.

DESCRIPTION: The User can click the SAVE and CANCEL buttons in the Account form as shown in Fig.29.

DEPENDENCY:

- The form finds no duplicate for ACCOUNT-NAME in Salesforce.
- The form may or may not be filled for the CANCEL button to function.

Functional Requirement 49

TITLE: The SAVE button in the Account form when the form is incomplete.

DESCRIPTION: If the User clicks the SAVE button without completing it then the record in the Accounts page appears with a “!” mark as shown in Fig.7b.

DEPENDENCY:

- The form is incomplete.
- At least one field in the form was filled by the User.

Functional Requirement 50

TITLE: The SAVE button in the Account form when the form is complete.

DESCRIPTION: If the User clicks the SAVE button after filling the form then the record in the Accounts page appears as shown in Fig.7a without the “!” form in the record after exporting it to Salesforce along with the associated contacts.

DEPENDENCY:

- Contacts were placed for that Account.

Functional Requirement 51

TITLE: The Sessions page.

DESCRIPTION: The Admin/User can view the current Session details AND/OR the previous Session details in the Sessions page.

DEPENDENCY:

- Admin added a User to a Session.

Functional Requirement 52

TITLE: The Admin creates a Session.

DESCRIPTION: By clicking the ADD NEW button and filling the form the Admin can create a Session as shown in Fig.12

DEPENDENCY:

- The Admin fills the form.
- The Admin Saves the form.
- Only the Admin can view this form and button.

Functional Requirement 53

TITLE: The Admin adds a User and a Campaign to a Session.

DESCRIPTION: The Admin must add at least one User to the Session and a Campaign to initiate a Session.

DEPENDENCY:

- The Admin creates a User before adding it to a Session.
- The Admin places a Campaign in Salesforce for the Session.

Functional Requirement 54

TITLE: The User can extend their time in a Session.

DESCRIPTION: If a User requires more time to complete a Session then the User makes a request to the Admin as shown in Fig.13. The User is prompted to confirm their decision with a pop-up as shown in Fig.14. with a YES and NO buttons. The YES button confirms the decision to proceed with and NO does not.

DEPENDENCY:

- The User makes the request within the time frame fixed by the Admin for that Session.

Functional Requirement 55

TITLE: The maximum number of days a User can request at a time for a Session.

DESCRIPTION: If the User makes a request for additional time, the maximum number of days a User is allowed to have at a time is nine. If the User needs more, then the User may request for additional days again.

DEPENDENCY: Functional Requirement 53

Functional Requirement 56

TITLE: Users/Admin can view all the exports to Salesforce in the EXPORTS page.

DESCRIPTION: The number of successful exports to Salesforce for every Session or Prospection including the number of users involved is documented as shown in Fig.16.

DEPENDENCY:

- No filtration criterion is set.

Functional Requirement 57

TITLE: The name of an Export should be of syntax: SESSION NAME_SESSION-START-DATE, else it should be GENERAL PROSPECTION.

DESCRIPTION: If a Session is named Bakkersvak-2017 and it starts on March 5th according to the START-DATE set by the Admin, then the Exports from that Session will be named as "Bakkersvak-2017_March 5th" as shown in Fig.16. If the Export does not belong to a Session, then the Export should be named GENERAL PROSPECTION.

DEPENDENCY:

- The Exports are not from other Sessions or from the Prospection.

Functional Requirement 58

TITLE: The User/Admin can filter the Exports based on certain criteria.

DESCRIPTION: The User/Admin can filter the Exports based on TYPE, USER and DATE as shown in Fig.17a.

DEPENDENCY: The Exports were not deleted by the Admin.

Functional Requirement 59

TITLE: The TYPE criteria to filter Exports.

DESCRIPTION: By using these criteria, the User/Admin filter the Exports on being either a SESSION or PROSPECTION.

DEPENDENCY: The User and Admin can select only one of the radio buttons corresponding to the criterion SESSION and PROSPECTION respectively to deliver the User/Admin requested result.

Functional Requirement 60

TITLE: The USER criteria to filter Exports.

DESCRIPTION: The User and Admin can filter the Exports based on which User exported them to Salesforce by check boxing that User as shown in Fig.17b.

DEPENDENCY: The User should have exported at least one Account with Contact(s).

Functional Requirement 61

TITLE: The DATE criteria to filter Exports.

DESCRIPTION: The User and Admin can filter the Exports based on its date from NCRM as shown in Fig.17b. The filter options available are as follows:

- TODAY: For all Users that created Exports on the day of filtering.
- YESTERDAY: For all Users that created Exports on the day before the day of filtering.
- LAST 7 DAYS: For all Users that created Exports on the seven days before the day of filtering.
- LAST 30 DAYS: For all Users that created Exports on the thirty days before the day of filtering.
- THIS MONTH: For all Users that created Exports on the month of filtering.
- LAST MONTH: For all Users that created Exports on the month before the month of filtering.
- CUSTOM RANGE: For all Users that created Exports between dates of User and Admin choice.

DEPENDENCY: Each Export sent by the User and Admin should have a time-stamp of when it enters Salesforce.

Functional Requirement 62

TITLE: The Admin creates a User.

DESCRIPTION: By clicking the ADD NEW button as shown in Fig.18 and filling the form as shown in Fig.19, the Admin can create a User.

DEPENDENCY:

- The User should have a Salesforce account.
- The application should integrate with the company Salesforce.
- The Admin completes the USERS form.

Functional Requirement 63

TITLE: The SPECIALIZATION drop-down menu in the USERS form.

DESCRIPTION: The Admin can select which specialization the User has from the following options in the drop-down menu:

- Print & Apply
- Printer
- Internal Sales
- General Sales

DEPENDENCY: The application should integrate with the company Salesforce.

Functional Requirement 64

TITLE: The Print & Apply option in the SPECIALIZATION drop-down menu.

DESCRIPTION: The Print & Apply option maps the User to the ‘P&A Rep.’ object in the ‘Accounts’ object in Salesforce. For each Account created by that User, it shall fill the ‘P&A Rep.’ object with that User.

DEPENDENCY:

- Functional Requirement 62
- The application should integrate with the company Salesforce.

Functional Requirement 65

TITLE: The Printer option in the SPECIALIZATION drop-down menu.

DESCRIPTION: The Printer option maps the User to the 'Printer Rep.' object in the 'Accounts' object in Salesforce. For each Account created by that User, it shall fill the 'Printer Rep.' object with that User.

DEPENDENCY:

- Functional Requirement 62
- The application should integrate with the company Salesforce.

Functional Requirement 66

TITLE: The Internal Sales option in the SPECIALIZATION drop-down menu.

DESCRIPTION: The Internal Sales option maps the User to the 'Internal Sales' object in the 'Accounts' object in Salesforce. For each Account created by that User, it shall fill the 'Internal Sales' object with that User.

DEPENDENCY:

- Functional Requirement 62
- The application should integrate with the company Salesforce.

Functional Requirement 67

TITLE: The General Sales option in the SPECIALIZATION drop-down menu.

DESCRIPTION: The General Sales option maps the User to the 'Owner' object in the 'Accounts' object in Salesforce.

DEPENDENCY:

- Functional Requirement 62

Functional Requirement 68

TITLE: The Admin can make another user into an Admin.

DESCRIPTION: By clicking the "" icon, the Admin is prompted with a pop-up message as shown in Fig.22, with the YES and NO buttons on it. Clicking YES will turn the selected User into an Admin and the current Admin will turn into a User.

DEPENDENCY: The Admin selects the User by check boxing, before clicking the ""icon.

Functional Requirement 69

TITLE: The OVERTIME page.

DESCRIPTION: All Over-time requests made by a User are available for approval from the Admin as shown in the Fig.24. for a Session. The Admin is provided with following details about each OVERTIME request from a User:

- USER-NAME
- SESSION NAME
- OVERTIME IN DAYS
- ORIGINAL DEADLINE
- EXTENSION DEADLINE

If there are no requests from any User then the OVERTIME page is shown as in Fig.23.

DEPENDENCY:

- Functional Requirement 53

Functional Requirement 70

TITLE: The APPROVE button in the OVERTIME page.

DESCRIPTION: The Admin can approve a User's request for overtime by clicking the APPROVE button as shown in Fig.24.

DEPENDENCY: The Admin check-boxes the User's request before clicking the APPROVE button.

Functional Requirement 71

TITLE: The THRASH button represented by " " icon.

DESCRIPTION: The User/Admin can delete a record by clicking the THRASH button.

DEPENDENCY:

- The record has been CHECKBOXED.

Functional Requirement 72

TITLE: The EDIT button represented by " " icon.

DESCRIPTION: The User/Admin can edit a form entry by clicking the EDIT button.

DEPENDENCY:

- The record is CHECKBOXED.

Functional Requirement 73

TITLE: The SEARCH button represented by " " icon.

DESCRIPTION: The User/Admin can search for a record in any of the pages with the SEARCH icon by entering a text string.

DEPENDENCY:

- The record of such string text should exist.

Functional Requirement 74

TITLE: The SEARCH bar shows the search result.

DESCRIPTION: If the User/Admin searches for a record and is found then it is shown in the SEARCH bar next to the SEARCH icon, otherwise it shows the message "NO RESULTS FOUND"

DEPENDENCY: Functional requirement 74

Functional Requirement 75

TITLE: The SETTINGS button is represented by “ ” icon.

DEPENDENCY: None

RESOURCES

1. Salesforcecom. (2017). *Salesforcecom*. Retrieved 28 March, 2017, from https://help.salesforce.com/articleView?id=integrate_what_is_api.htm