

Software Requirements Specification (SRS) Document

Project Title: Boostr sales assistant bot

Project Number: SSAD 47

Team Number:35

Members:

1. Buggana Sathvik Sanjeev
2. Mugdha Yashodhan Abhyankar
3. Ayila Koushik Sai Achyuth

Client: Chirag Chowdhary, NebulaeSoft

Brief problem statement

The goal of this project is to provide a sales assistant to analyse and drive profitable behaviour among prospective companies.

To profitably analyze statistics for the dashboard of a user and also to serve as a digital assistant for many platforms. It augments and accelerates profitable behavior across your sales network, using context-relevant insight and access.

System requirements

~ 2GB free RAM and i5/i7 processor for testing/development

Database Manipulation: MongoDB-Compass

API Tester: Postman, Swagger

Programming language: JavaScript (NodeJS, ExpressJS)

Users profile

There are four types of users (Sales, Partner Head, Vendor, Admin).

The tool is being built for those with minimal and basic knowledge of computers. No specific expertise in technology would be required.

Feature requirements

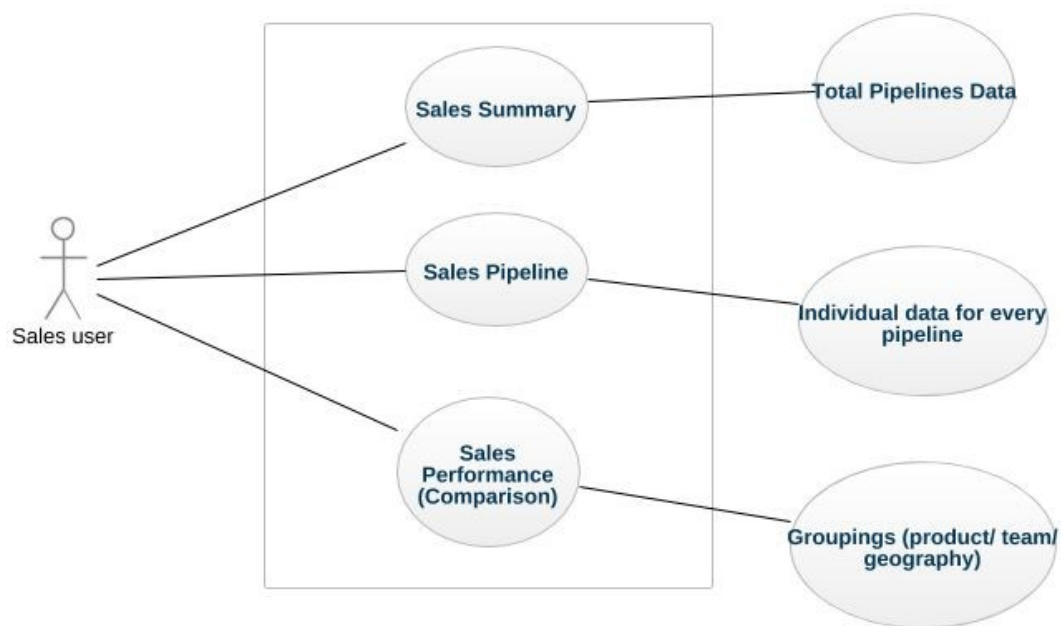
A dashboard API is to be built for each of the user.

A sales pipeline depicts the individual steps sales-people take through the different stages: Lead, Prospect, Opportunity, Negotiation, also some more stages depending on the organisations the user has. until the lead is removed from pipeline ("won" or "lost" state). By default it would be in an Open state.

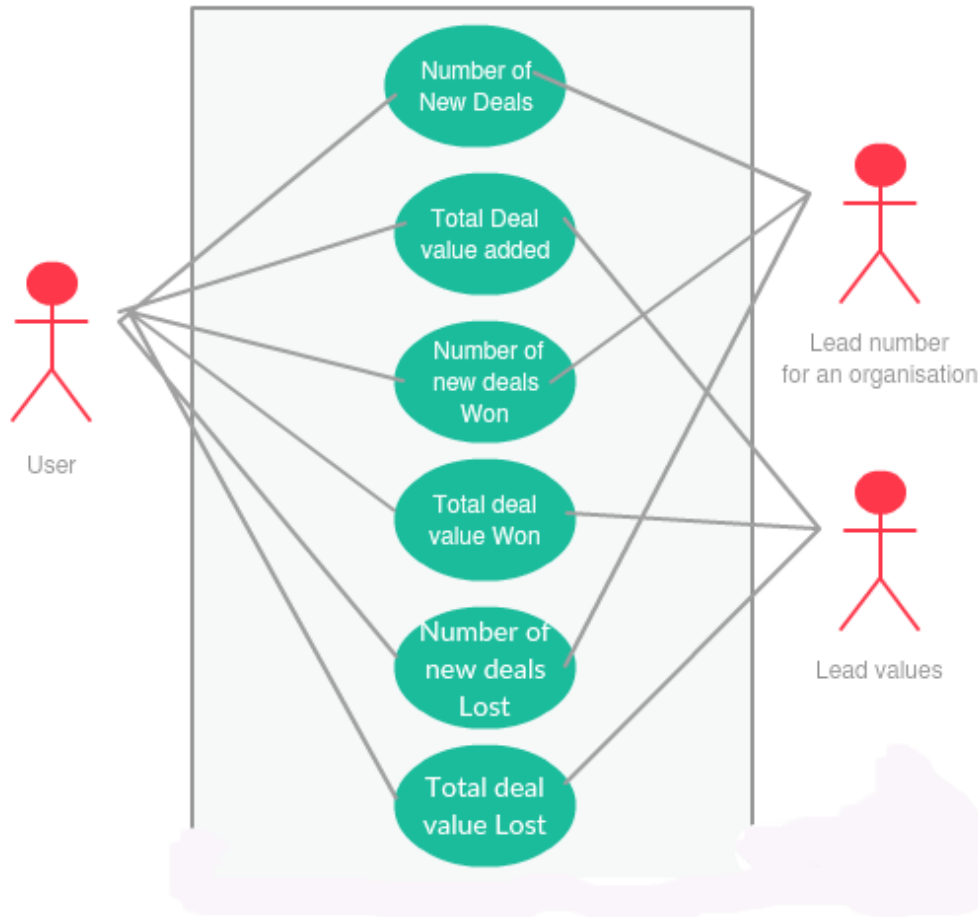
All the functionalities are implemented for a organisation, which is linked to a user of any type.

No:	Functionality:	Description:	Release:
1.	Sales Summary Model	<p>The Admin (Sales Manager) needs to view the overall company level stats. The model needs to be drafted and the corresponding analysis is designed</p> <ul style="list-style-type: none"> • Total number of deals currently in pipeline • Total Deal Value currently in pipeline • Average Time to closure, in days, of all deals closed 	R1
2.	Sales Summary API	Add a wrapper function to co-ordinate the above with a duration. Scheduling the functions per day using cronJob. Integration with existing code base	R2
3.	Sales Pipeline Model and Functions Only for stageId	<p>->Retrieve computations for every stage ->Pipeline stats:(StageId as a param, OrgId has lists of StageID's TODO for R2)</p> <ul style="list-style-type: none"> • Stage name • List of deals with <ul style="list-style-type: none"> ◦ Deal Name ◦ Deal Value • Total number of deals currently in that stage • Total deal value currently in that stage • Percentage of total added deals reaching that stage • Percentage of deals that move on to the next stage • Percentage of deals that are lost in that stage 	R1
4.	Functions for OrgId and Sales Pipeline API	<p>-> User comes into the picture now as we take his/her organisationId and list of stageId's attached to it doing above computations ->Add a wrapper function to co-ordinate the above with a duration. Scheduling the functions per day using cronJob. ->Using Morgan to log when the functions start and end. Integration with existing code base.</p>	R2
5.	Sales Performance Models	<p>The User views deals in terms of value/ number of deals. Draft models and use existing models to extract data from database. The details are for:</p> <ul style="list-style-type: none"> • Deals that are currently in the pipeline • Deals that were Won • Deals that were Lost • Deals that were Added • Average time to closure • Analysis of Wins and Losses 	R1
6.	Sales Performance API	Add a wrapper function to co-ordinate the above with a duration of 1 month, functions need to use Mongoose as the ODM to query	R2

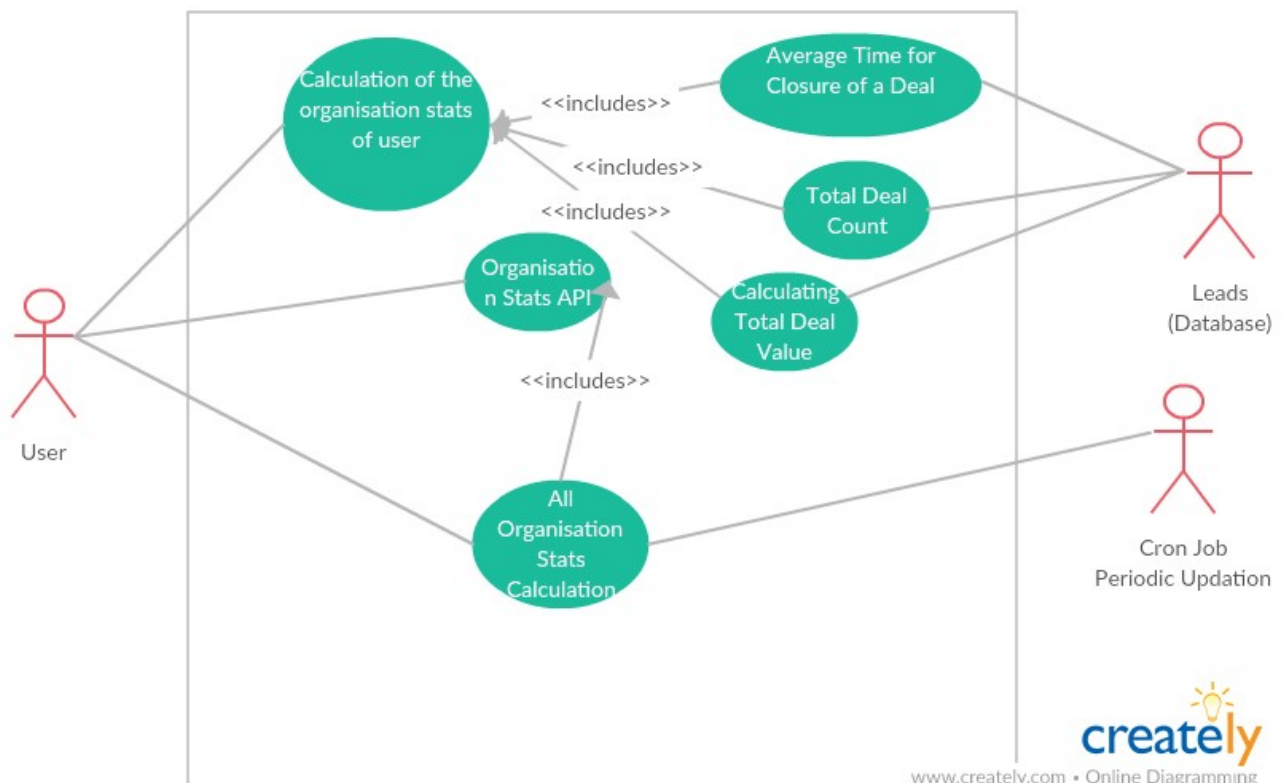
		existing models. Scheduling the functions per day using cronJob. Integration with existing code base.	
7	Overall Stats Model	<p>The Sales Manager needs to view stats of an organisation. A model is required to store the following data</p> <ul style="list-style-type: none"> • Total Deal Count of an organisation • Total Deal Value of an organisation's leads • Average closure time for a Deal 	R1
8	Calculating the overall stats of an organisation	Each user/ admin has an organisation has its corresponding overall stats which are to be calculated from the set of all leads present in the database	R1
9	Organisation's overall stats api	Each user, when logged in, needs to see his/her's organisation stats in the dashboard for which an API is needed to calculate and retrieve the data.	R1
10	Periodic Calculation of overall stats	The overall stats of each organisation need to be updated daily to ensure that these stats remain updated.	R2



Sales Performance



Overall Stats



Sales Pipeline Stats

