# DASS
# Project Report
# Build a prospective list of customers based on Ideal Customer Profile
## Team 25

Astitva Gupta - 2018101085
Sachin Kumar Danisetty - 2018101108
Trusha Sakharkar - 2018101093
Vivek Pamnani - 2018111032

## Introduction

This product is for B2B businesses and other individuals wanting to sell their product.

In the business world, the 80/20 rule suggests that for a business, 80% of their net income is derived from 20% from their overall consumer base. It therefore makes sense that, businesses should focus on delivering to that 20% of the consumers. A way to approach this problem is narrowing the potential consumer base by a certain screening process.

Ideal Customer Profile (ICP) is essentially the filter of that screening process.

According to the 80/20 rule, it becomes apparent that spending time and resources on approaching each potential consumer would prove cost inefficient. Thus, by narrowing that list using an ICP, a business would ideally minimize costs whilst sustaining a good income.

To achieve this, a company could hire an employee for the screening process. However, for large businesses this could easily turn out to be a time-consuming process.
A better yet solution would be automating it. That is precisely our **target**.

## Software Requirements Document (SRS)

*Description:*
Software system to crawl the web and profile prospective customer businesses fitting a given Ideal Customer Profile. Apart from gathering raw data from the web, the software system would clip and summarize relevant articles while attempting to appropriately categorize them. The resulting database can be navigated/searched using a UI.

*User Profile:*
The software system is of utility to individuals, B2B businesses looking for prospective customers to effectively mitigate the marketing costs while boosting sales and to develop profitable relationships.

*Functional Requirements:*

| No. | Use Case | Description | Release |
|---|---|---|---|
| 1. | User Authentication | User can log into the system via Google Auth. | R1 |
| 2. | Search | User can input a search phrase along with attributes/filters and initiate the search. | R1 |
| 3. | View Search History | User can view the list of their previous searches (hyperlinked to results) and opt for a new search. | R1 |
| 4. | Profile | User can view their personal information and opt to view their search history. | R1 |
| 5. | Logout | User can log out of the system ending the session. | R1 |
| 6. | Show Results | User can view the results of a chosen search and choose sorting parameters as desired. | R1 |
| 5. | View List of Contacts/Founders | Users can view the people to be contacted for communication with the prospective customer. | R2 |
| 6. | Company Knowledge Base | Users can view the list of interesting news articles, announcements, etc. about customer companies. | R2 |
| 7. | Startup Profile | User can view the complete details of a search result. | R2 |

*Technologies:*
MERN Stack for the WebApp.

- Database: MongoDB
- Backend: NodeJS
- Frontend: ReactJS
- Web Framework: ExpressJS

Python library Scrapy: To gather information from sources which do not provide an API.

## Design Document

*Architectural Design:*

- Server: To get and set values for other classes, provide functionality.
- User: Authentication module to verify Google SSO and get user's email.
- Search: Module to handle the search query processing. Returns results in JSON format.
- UI: The UI module.

*User Interface:*

- Login via Google SSO by clicking the button on the welcome page.
- Search History: The page allows the user to review past searches or initiate a new search.
- Search: User can enter desired attributes to find a fitting startup profile.
- Results: A complete description of a result is displayed on clicking the result entry on the results page.

*Design Rationale:*
Initially, the software system would display all the details on the results page itself. Therefore, the startup specific information was moved to a separate profile page.
Loading animation was introduced to make the system feel more responsive (since, a search might take a long time).

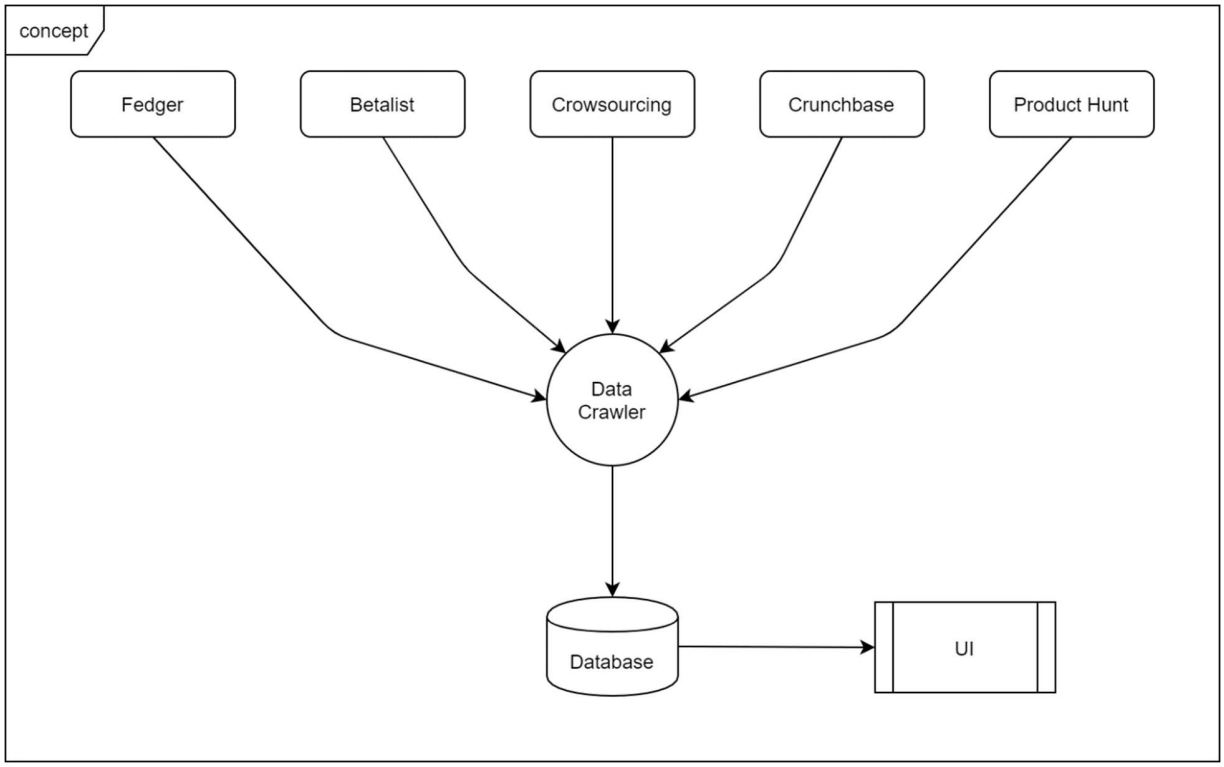
## Project Concept

*Description:*
Software system to crawl the web and profile prospective customer businesses fitting a given Ideal Customer Profile. Apart from gathering raw data from the web, the software system would clip and summarize relevant articles while attempting to appropriately categorize them. The resulting database can be navigated/searched using a UI.

*User Profile:*
The system is of utility to users fitting the following descriptions:

- Entrepreneurs:
  Person or group of people who are setting up a business or businesses.
- Sales Division:
  Activities related to selling or services sold in a targeted time period. These people take care where the company should sell its products and is it profitable to do business with another company.
- Investors:
  Person or other entity who commits capital with the expectation of receiving financial returns or to gain an advantage.
- Market Researchers:
  People that help to strategize the company's projects and maintain a steady base of customers for the business.

*Concept:*




## Project Plan

*Description:*
Software system to crawl the web and profile prospective customer businesses fitting a given Ideal Customer Profile. Apart from gathering raw data from the web, the software system would clip and summarize relevant articles while attempting to appropriately categorize them. The resulting database can be navigated/searched using a UI.

*Development Environment:*
Development Environment: Visual Studio Code
Programming Languages:

- HTML
- CSS
- JavaScript
- Python

API / Tools:

- ReactJS
- Scrapy
- Crunchbase
- Twitter

*Milestone Schedule:*

| No. | Milestone | Description | Due Date | Release | Deliverable |
|---|---|---|---|---|---|
| 1. | Requirements Gathering | - | 21-01-20 | - | No |
| 2. | Requirements Analysis | Part of feasibility study to classify requirements in and out of scope. | 28-01-20 | - | No |
| 3. | Generating SRS | Describes the functional requirements, non-functional requirements, and goals. | 29-01-20 | - | No |
| 4. | Design Document | Describes the design of a system for software development mentioning the 'how' and 'why'. | 01-02-20 | - | No |
| 5. | Project Plan Document | Describes the outline and features implemented (or to be) of the project. | 01-02-20 | - | No |
| 6. | User Authentication | User can log into the system via Google Auth. | 14-02-20 | R1 | Yes |
| 7. | Search | User can input a search phrase along with attributes/filters and initiate the search. | 14-02-20 | R1 | Yes |
| 8. | View Search History | User can view the list of their previous searches (hyperlinked to results) and opt for a new search. | 29-02-20 | R1 | Yes |
| 9. | Profile | User can view their personal information and opt to view their search history. | 29-02-20 | R1 | Yes |
| 10. | Logout | User can log out of the system ending the session. | 14-02-20 | R1 | Yes |
| 11. | Show Results | User can view the results of a chosen search and choose sorting parameters as desired. | 29-02-20 | R1 | Yes |
| 12. | Updating UI | Update the UI to support:<br>1. New page to display all information related to a search.<br>2. Toggle a search to display some additional information. | 14-03-20 | R2 | No |

| | | | | | |
|---|---|---|---|---|---|
| 13. | Incorporating larger source domains. | Larger number of source domains for data. | 23-03-20 | R2 | No |
| 14. | Email Notification | Sends an email notification on completion of a search task. | 23-03-20 | R2 | Yes |
| 15. | Source Links | Redirects to the source page for a selected result. | 11-04-20 | R2 | Yes |
| 16. | News | Collect links to relevant news articles. | 11-04-20 | R2 | Yes |
| 17. | Startup Profile | Display the details of a search result in a separate page along with additional information. | 11-04-20 | R2 | Yes |
| 18. | View List of Contacts/Founders | Users can view the people to be contacted for communication with the prospective customer. | 11-04-20 | R2 | Yes |
| 19. | Company Knowledge Base | Users can view the list of interesting news articles, announcements, etc. about customer companies. | 11-04-20 | R2 | Yes |

## Test Strategy

*Scope:*
Review: The document will be reviewed by both client and the team.
Approval: Client and Mentor of the team will approve the document.

*Test Approach:*
The product is tested on four levels:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

*Testing Tools:*

- Postman: To test backend components.
- React Dev Tool: To test frontend components.
- Passport: To test SSO login via Google. Serves as a middleware.
- Mongo Compass: To test the database.

## Test Analysis

*Unit Testing:*
Unit Testing is a level of software testing where individual units/ components of a software are tested. It is the first level of software testing.
Example,
The domain attribute was supplied other than the allowed list of domains for the search form unit.
Expected Outcome: Search failed due to incorrect input, input again.
Observed Outcome: Search successful, results empty.

*Integration Testing:*
Integration Testing is a level of software testing where individual units are combined and tested as a group. It is the second level of software testing.
Example,
The SSO login unit was developed first alongside the search feature which did not require a user to be logged in. Once each unit was verified working, both were integrated and tested if the searches were user specific. Initially, a search without a user logged in would be stored causing NULL exceptions.
Expected Outcome: Search failed due to unauthenticated user.
Observed Outcome: Search proceeds with empty results.

*System Testing:*
System Testing is a level of software testing where a complete and integrated software is tested. It is the third level of software testing.
Example,
In certain test cases, some UI elements were misplaced.
Expected Outcome: The UX is consistent.
Observed Outcome: The information was difficult to infer.

*Acceptance Testing:*
Acceptance Testing is a level of software testing where a system is tested for acceptability. It is the last level of software testing.
Example,
Some NULL exceptions were not taken care of when searching for startups. E.g., the region/country attribute was left empty or a non-existent profile link (NULL value) was displayed.
Expected Outcome: Profile is not displayed.
Observer Outcome: Profile button redirects to a blank page.

## Technologies Used

MERN Stack for the WebApp.

- Database: MongoDB
- Backend: NodeJS
- Frontend: ReactJS
- Web Framework: ExpressJS

Python library Scrapy: To gather information from sources which do not provide an API.

## Challenges Faced

Some of the major challenges faced were:

- Most of the open data sources do not allow bots to crawl through their domain.
- We had to select suitable resources and modify bots to get them allowed to crawl through their domains.
- We have data with different attributes, and they are to be integrated with a single database collection. We have to design the database schema very cautiously.
- While crawling the URLs, we faced some syntax problems like spaces in an URL, which we solved by using scripts.
- Setting up the CSS. The CSS code acts more in a universal nature across components making it harder to customize each component.
- Setting up the background to be different for all the webpages and changing the elements.
- Scaling of elements in components.
- After the pandemic hit, establishing communication across the team and collaboration was difficult.

## Crucial Learnings

- **Agile/Scrum:**
  The project was divided into sprints which helped us in being organized and also simulated the real-world situations.
- **Team communication:**
  Most of the components in the project are interrelated so team communication was a crucial part of the project.
- **Git:**
  Git being the most used tool in the real-world the whole team learned how to use the tool to its best extent.
- **Corporate:**
  Through this project we had the opportunity to learn various topics in the business world. The 80/20 rule, about technographic data, startup development, B2B and B2B businesses, etc.