1. Introduction

This Document has the proper documentation of the minor project named as **huskdata** . the document refers to all the technical and non technical stuff related to the project . the document will answer questions such as why the project is required ?(Purpose),What are the future scopes that project may attain ?(Scope) , how the project is going to be implemented (SRS).

1.1 Purpose

The project is a data science project which will deal with all the data preparation techniques and making it easier to get to it . the data is so precious but the moment it asks for its reference it becomes useful . the data when gets a reference it can be converted into various forms of data . the data is very useful in various situations say fields such as earthquake detection system , medical tech devices etc. the project is prepared for the Data preparations of several data files so that the person analyzing the critical data may not get the hustle to prepare the data for the statistical analysis

Our goal :

To make a product that makes data preparations without losing a single record as per data set has its own properties so making a system irrespective of the properties .

1.2 Scope

The project is basically a data preparation + data virtualization tool which will make the bar graphs and all respective charts that are relevant to the data.
The visualization is important so that it can be understood by the person requesting prepared data

The project is useful for a data analyst (First User). So the data preparation tool will stand with the other ones .

1.3 Definitions, Acronyms and Abbreviations

Data preparation tool:

Data preparation tools are a new class of software products designed to enable business analysts and data scientists to bypass data warehouses to perform some data integration and data preparation themselves before analysis. Data preparation tools can search for and access data throughout an organization, combine it with other, external data sets and do data cleansing and conversions as required before feeding the data back into business intelligence systems for analysis.

These emerging tools use machine learning under the hood so that they can iterate and learn where to find insights in data sets, without being explicitly programmed to do so.

Data visualization tool:

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Data Analyst:
Data analysts translate numbers into plain English Every business collects data, whether it's sales figures, market research, logistics, or transportation costs. A data analyst's job is to take that data and use it to help companies make better business decisions. This could mean figuring out how to price new materials for the market, how to reduce transportation costs, solve issues that cost the company money, or determine how many people should be working on Saturdays.There are many different types of data analysts in the field, including operations analysts, marketing analysts, financial analysts, etc.

1.4 References

- https://www.snagajob.com/job-descriptions/data-analyst/
- https://www.rasmussen.edu/degrees/technology/blog/what-does-a-data-analyst-do/
- https://www.tableau.com/learn/articles/data-visualization
- https://www.trustradius.com/data-preparation
- https://www.sciencedirect.com/science/article/abs/pii/0019103579900095
- https://scholar.google.com/
- https://pandas.pydata.org
- https://en.wikipedia.org/wiki/Data_analysis

1.5 Overview

This is a working document and, as such, is subject to change. In its initial form, it is incomplete by definition, and will require continuing refinement. Requirements may be modified and additional requirements may be added as development progresses and the system description becomes more refined. This information will serve as a framework for the current definition and future evolution of the **HUSK DATA**.

Overall Description

The Project is all about the data preparation and data virtualization to make the work of an analyst easier . the project will collect the headers from data sources so that the algorithm know that the data has what kind of reference . the project will convert the data into categorical data and at last to a statistical data .at last the project will show all the visual graphs to the user so that he/she must be able to verify that the data produced is a valid data. and the operations performed on the data are correct . if the user has given a negative feedback that the data has not been processed correctly so an algorithm which takes a bit more time will be launched so that the data can be wrangled more accurately. After the user has verified the data the csv file will be exported and the user will be able to download the file from the interface which will be provided to him .

2..1 Product perspective

The project is intended for the help of an analyst and a developer who is in need of data preprocessor to make his/her work more faster so that he can focus on some other stuff he had to do . this kind of situations are very offending and harsh to be faced when deadline is there for a bigger project so the app will be making sure that the user gets his/ her data as fast as possible .

System Interface :

The System interface purely depend on the architecture so at last we decided to choose the micro service Architecture to deploy the application . the architecture will make sure that if a bug happens in one app it is not going to affect the other system services or say apps .

User Interface :

The user interface is avery crucial phase of the project creation so making a user interface that is simple as well as attractive so to achieve the task we are using the web interface technologies i.e. HTML, CSS, JavaScript.

Hardware interface :

Server side:

The server will be listening to application at port 80 which is the default port .

Client side:

The client side hardware interface needs a device with an internet connection and a 2 gigs of ram .

Communication interfaces:

HTTP will be used as a communication interface between the client and the application.

Memory Constraints :

The memory constraints are necessary as per the data file is being read and write simultaneously.

Software Interfaces :

Server side:

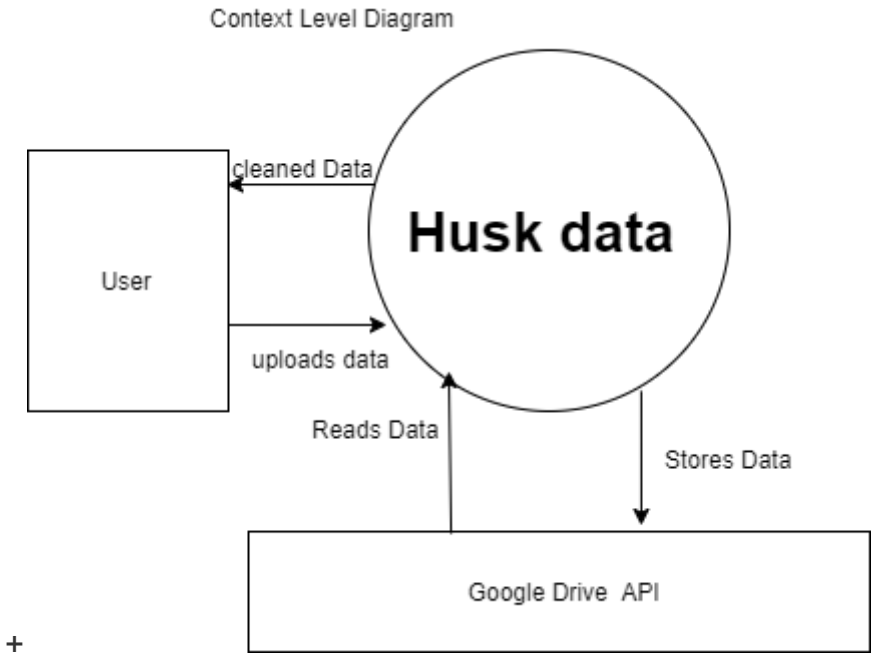
Softwares such as a python interpreter and all the required libraries must be there to make sure the project is functioning properly and the apache server is required to run the application.

Client Side:

A web browser is sufficient to get the application's user interface .

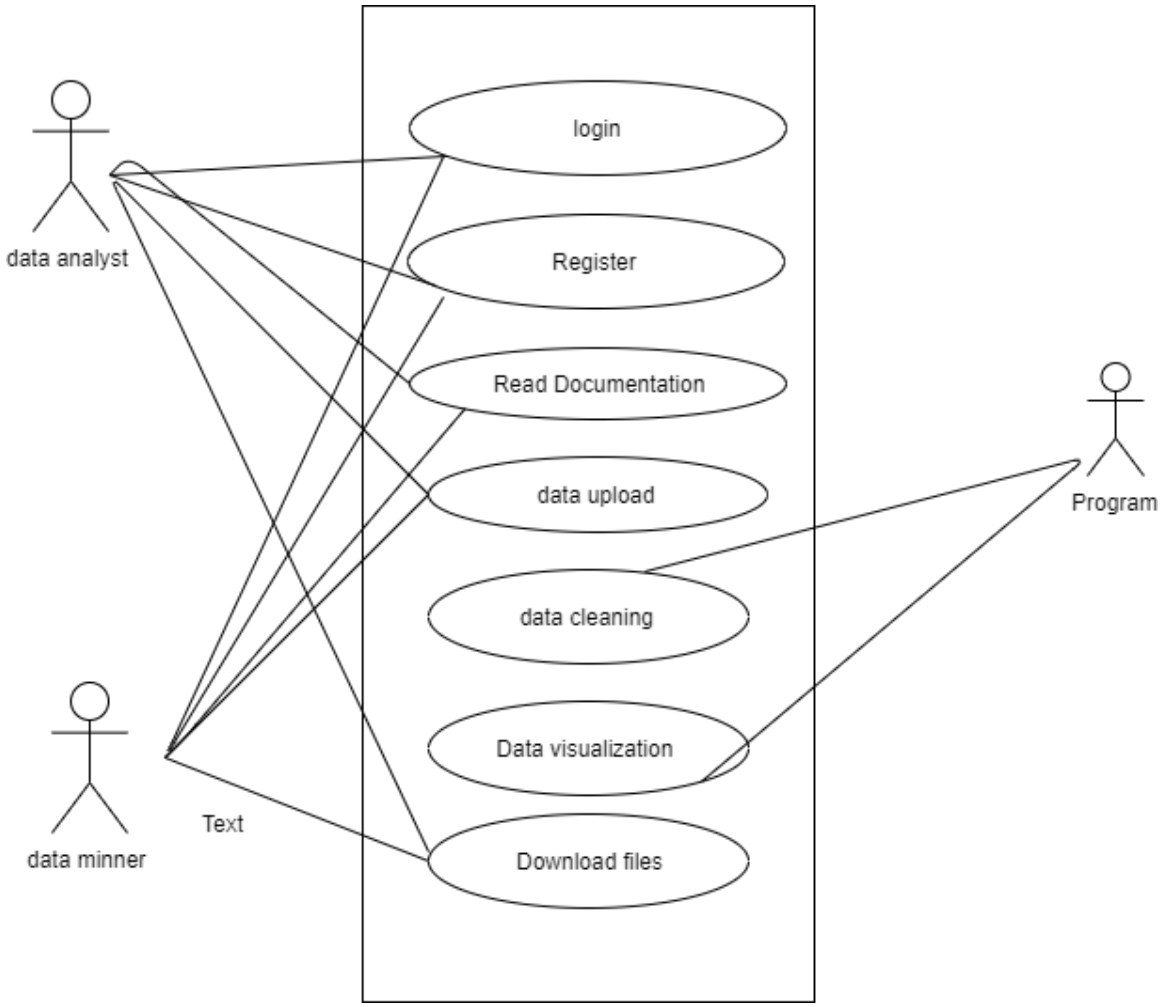
2.2 Product Functions :

2.2.1 Context Diagram :



+

2.2.2 Use Case Diagrams :



2.2.3 Use Case Description/Introductions:

2.2.3.1 Groups :

The groups that exist in our project are :

1. **Admin** : The super power holder in the project . the role of the admin is to maintain the project sources and the data source that is the DataBase.
2. **User** : The User of the project is eligible for registration as well as login and uploading the data files that must be in csv or json format .

2.2.3.2 **Data Cleaning** :

The Data cleaning is the module in the project that deals with the main tasks related to the project . the data cleaning is very important as the data without cleaning may result in various errors during analysis and mining it . and due to the critical nature of data , the data gets more disturbed while performing the operations on it . the data cleaning procedure has certain steps .

Index:

Data quality

Frankly speaking, I couldn't find a better explanation for the quality criteria other than the one on Wikipedia. So, I am going to summarize it here.

Validity

The degree to which the data conform to defined business rules or constraints.

- ***Data-Type Constraints:*** values in a particular column must be of a particular data type, e.g., boolean, numeric, date, etc.
- ***Range Constraints:*** typically, numbers or dates should fall within a certain range.
- ***Mandatory Constraints****:* certain columns cannot be empty.
- ***Unique Constraints:*** a field, or a combination of fields, must be unique across a dataset.
- ***Set-Membership constraints*****:** values of a column come from a set of discrete values, e.g. enum values. For example, a person's gender may be male or female.
- ***Foreign-key constraints*****:** as in relational databases, a foreign key column can't have a value that does not exist in the referenced primary key.
- **Regular expression patterns:** text fields that have to be in a certain pattern. For example, phone numbers may be required to have the pattern (999) 999–9999.
- **Cross-field validation:** certain conditions that span across multiple fields must hold. For example, a patient's date of discharge from the hospital cannot be earlier than the date of admission.

Accuracy

The degree to which the data is close to the true values.

While defining all possible valid values allows invalid values to be easily spotted, it does not mean that they are accurate.

A *valid* street address might not actually exist. A *valid* person's eye colour, say blue, might be valid, but not true (doesn't represent the reality).

Another thing to note is the difference between accuracy and precision. Saying that you live on the earth is actually true. But, not precise. Where on the earth?. Saying that you live at a particular street address is more precise.

Completeness

The degree to which all required data is known.

Missing data is going to happen for various reasons. One can mitigate this problem by questioning the original source if possible, say re-interviewing the subject.

Chances are, the subject is either going to give a different answer or will be hard to reach again.

Consistency

The degree to which the data is consistent, within the same data set or across multiple data sets.

Inconsistency occurs when two values in the data set contradict each other.

A valid age, say 10, mightn't match with the marital status, say divorced. A customer is recorded in two different tables with two different addresses.

Which one is true?.

Uniformity

The degree to which the data is specified using the same unit of measure.

The weight may be recorded either in pounds or kilos. The date might follow the USA format or European format. The currency is sometimes in USD and sometimes in YEN.

And so data must be converted to a single measure unit.

The workflow

The workflow is a sequence of three steps aiming at producing high-quality data and taking into account all the criteria we've talked about.

1. **Inspection:** Detect unexpected, incorrect, and inconsistent data.
2. **Cleaning:** Fix or remove the anomalies discovered.
3. **Verifying:** After cleaning, the results are inspected to verify correctness.
4. **Reporting:** A report about the changes made and the quality of the currently stored data is recorded.

What you see as a sequential process is, in fact, an iterative, endless process. One can go from verifying to inspection when new flaws are detected.

Inspection

Inspecting the data is time-consuming and requires using many methods for exploring the underlying data for error detection. Here are some of them:

Data profiling

A **summary statistics** about the data, called data profiling, is really helpful to give a general idea about the quality of the data.

For example, check whether a particular column conforms to particular standards or pattern. Is the data column recorded as a string or number?.

How many values are missing?. How many unique values in a column, and their distribution?. Is this data set is linked to or have a relationship with another?.

2.2.3.3 **Data Visualization :**

While these may be an integral part of visualizing data and a common baseline for many data graphics, the right visualization must be paired with the right set of information. Simple graphs are only the tip of the iceberg. There's a whole selection of visualization methods to present data in effective and interesting ways.

Common general types of data visualization:

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards

More specific examples of methods to visualize data:

- Area Chart
- Bar Chart
- Box-and-whisker Plots
- Bubble Cloud
- Bullet Graph
- Cartogram
- Circle View
- Dot Distribution Map
- Gantt Chart
- Heat Map
- Highlight Table
- Histogram
- Matrix
- Network
- Polar Area
- Radial Tree
- Scatter Plot (2D or 3D)
- Streamgraph
- Text Tables
- Timeline
- Treemap
- Wedge Stack Graph
- Word Cloud
- And any mix-and-match combination in a dashboard!

Specific Requirements

3.1 External Interface

3.1.1 Web Server :

- Apache will be used as webserver
- The user inputs data will bc collected through web server using html forms
- The Web Server executes the python as module of python retrieves the data if available
- The Web Server displays a html page as result to the end-user

3.1.2 Python - Django Application

The whole web Application is created and served using the python and django . all the user information required for login will be stored in the database and the user files will be stored in their respective google drives .

3.1.3 Databases :

MySQL :

The database will be treated as the main database as it is secure and can handle a lot data by default .

SQLite :

The Database will serve as a backup database to the project .

3.2 Functional Requirements :

3.2.1 Use Case Scenario :

User Registration

Purpose	User here registers himself into the database
User	here anyone can register himself
Input Data	Some personal information or a google account
Output Data	The user gets added into the database
Invariants	Profile table data and user information
Pre-conditions	User may not be registered into the website but through this he/she can .
Post - condition	If Information is provided it must be verified
Basic Flow	The data flow will be unidirectional from html to database
Alternative Flow	Checking all the formats and necessary details validation using javascripts and html and django

Business Rules	This allows user to create an account of themselves

Login

Purpose	User logs into the system
User	A user with existing credentials or an google account
Input Data	Username, password or google account
Output Data	The user gets the dashboard to upload his/her files
Invariants	Profile table data and user information
Pre-conditions	User is not logged into a profile , input is required and is necessary
Post - condition	If Information is provided it must be verified
Basic Flow	The data flow will be Bidirectional from html to database and from database to HTML
Alternative Flow	Validation of correct username and password
Business Rules	This allows user to get the dashboard for futher

Data Preparation

Purpose	User gets the data cleaned
User	Uploads a file
Input Data	User uploads a raw data file in supported formats
Output Data	The user ges cleaned data
Invariants	Data type specification and all other parameters
Pre-conditions	User must be logged on and give drive authorization
Post - condition	The cleaned data file is generated
Basic Flow	The data flow will be Bidirectional from drive to application and from application to drive

Alternative Flow	Matching patterns and cleaning the data according to it
Business Rules	This allows user to upload and download the data files

3.3 Performance Requirements :

The System must be scaled as the users increases in the database.

3.4 Logical Database Requirements :

All data will be saved in the database: user accounts and profiles, discussion data, messages etc. (except files which are stored on the disk.) The database allows concurrent access and will be kept consistent at all times, requiring a good database Design.

3.5 Design Constraints

1. The communication between the portal software and the database will be in SQL.
2. The page layout will be produced with HTML/CSS.
3. The product will be written in Python and Java(Android APP).
4. The output must be compatible with W3C XHTML 1.0
5. The source code must follow the coding conventions of Python
6. System administrators must have access to comprehensive documentation.

3.6 Software System Attributes

The software consists of the following elements:
1. The apache web server
2. The Python application
3. The MySQL and SQlite database
4. The database should remain consistent at all times in case of an error.\

3.6.1 Reliability

The reliability of the overall program depends on the reliability of the separate components.

3.6.2 Availability

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved with the MySQL server and saved by the administrator.

3.6.3 Security

1. Passwords will be saved encrypted in the database in order to ensure the user's Privacy.
2. The user's IP will be logged.
3. The system will be protected against vulnerabilities such as SQL injection

Attacks.

3.6.4 Maintainability

MySQL is used for maintaining the database and the Apache server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

3.6.5 Portability

The application is created on django and should be compatible with other systems. Apache, Python and MySQL programs are practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web browser should be able to use the features of the application.