Software Requirements Specification

Predictive Analysis of Open Govt. Data (OGD) Platform India (data.gov.in)

Made By:

Devendra Pratap Yadav

Jatin Garg

Sakshum Sharma

Simarjit Singh

Snehil Ameta

Table of Contents

1. Introduction

1.1 Purpose

This SRS covers the requirements of a web application that enables analytics and prediction based on OGD available on data.gov.in. This covers the front-end(web interface) as well as back-end(Web server, database, python programs) components of the application.

1.2 Document Conventions

The acronyms used throughout the document are listed below:
TBD - To Be Decided
OGD - Open Government Data

1.3 Intended Audience and Reading Suggestions

This document is intended for developers of the application, beta testers as well as users. The document describes the product scope, users and platform in Section 2. Section 3 describes external interface requirements for the product. A detailed description of product features are provided in section 4. Section 5 describes non-functional requirements of the product.
The document should be read in order of sections. Developers and testers must read all sections. Common users may give more priority to Section 2 and 4.

1.4 Product Scope

A large amount of statistical data is publicly available on data.gov.in. This data is generally accessible through file downloads only with no online viewing capability. The software will be able to provide users with ability to select, visualize and predict future values for various datasets.
This software can be of great utility for common citizens, small scale businesses, farmers and govt. resource allocation departments. The goal is to make the public data easily accessible to all with useful analysis capabilities.

1.5 References

OGD (Open Govt. Data) is used as the source of datasets and their description. The APIs associated with the datasets are used to fetch them and store in our database.

2. Overall Description

2.1 Product Perspective

Under the digital India initiative, statistical data has been publicly released on data.gov.in. We aim to develop a web-based application to utilize the aforementioned data and provide analytics and prediction services. We are interested in predicting useful parameters like prices of commodities, regional crop yield prediction, mortality rates, etc. which is valuable for common citizens as well as small businesses. This document describes the application with details of components and functions.

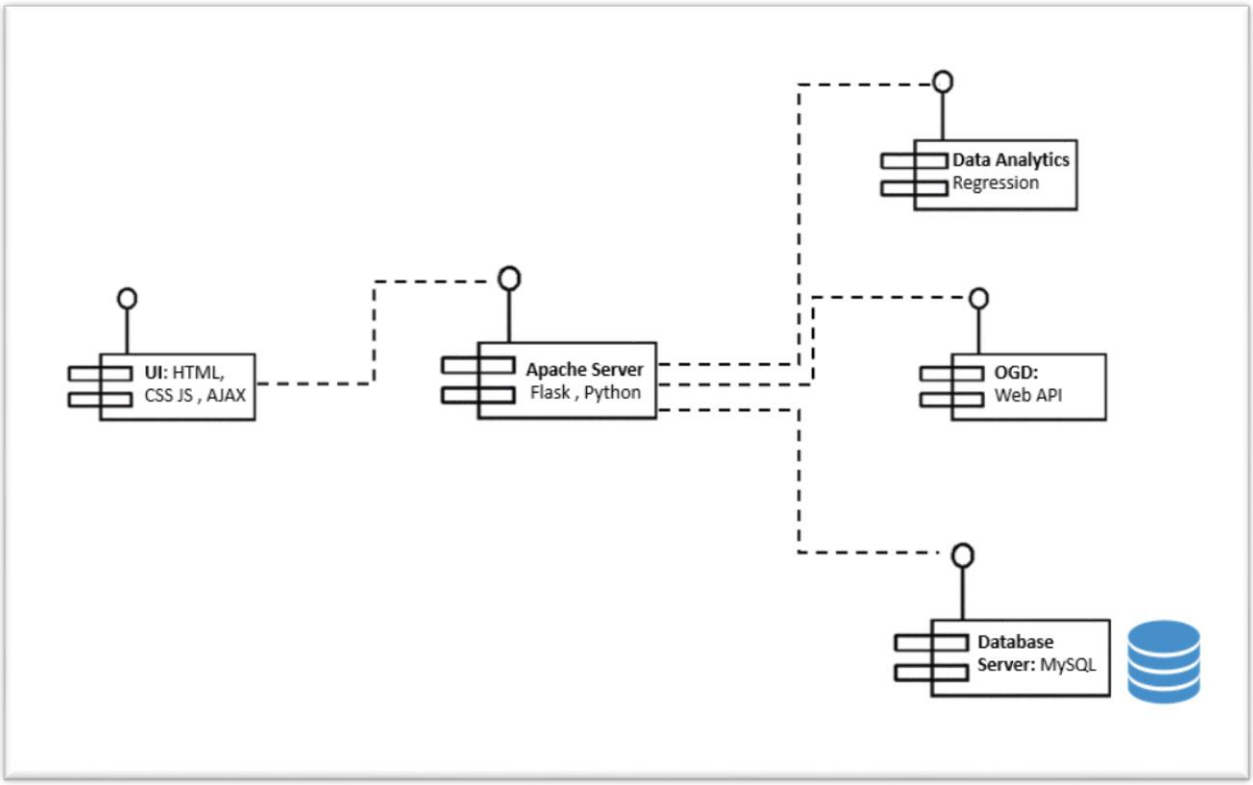


Figure 1 : Component Diagram

2.2 Product Functions

The application enables the user to select a dataset or its subset and view suitable statistics for it in graphical or tabular format. Based on the dataset used and parameters calculated the user should be able to predict future values of some parameters.
The broad functionality is summarized in the Interaction diagram.

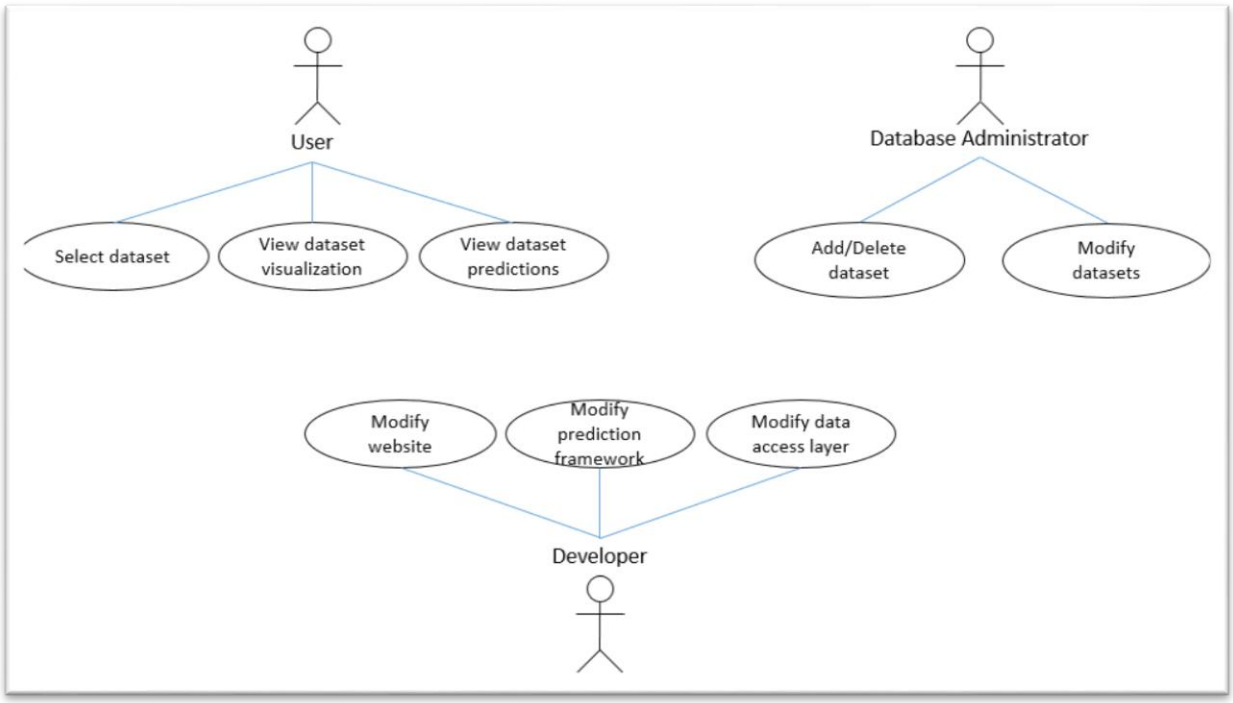


Figure 2 : Use Case Diagram

2.3 User Classes and Characteristics

This product will be very helpful for planning and strategizing based on the statistics that the model presents.
The application users can be categorized as:

- **Common citizens**: High priority.
  The application will provide informative view of OGD. Citizens can use the application to analyse past statistics and obtain relevant predictions.
- **Small business owners**: Moderate priority.
  Small business owners can use data visualizations and predictions for relevant domains, and make informed business choices.
- **Government offices**: Moderate priority.
  Government offices can use the application to get rough predictions of various national statistics; utilizing them in policy making and planning.
- **Application developers**: High priority.
  Developers of the application will have administrative access. They will be able to add/modify databases, web interface and test various application components.

2.4 Operating Environment *

The code should be able to run on any platform running python. This is something that will be updated over time, as and when the code comes close to its final stage.

The frontend web-based application can be operated on any system with a modern browser such as Firefox/Chrome over the Internet.
The backend will operate in a Linux environment. MySQL database and Python are pre-requisites.

2.5 Design and Implementation Constraints

The development of a generic analytics application for all databases is constrained by two factors. Firstly, downloading existing datasets is mostly manual since APIs are not available for all of them. Secondly, we need to clean and filter the data. The datasets are typically multivariate and often have several attributes as 'NULL'. To be able to train machine learning models we will need to get NULL-free data.
The application is also constrained by the availability of prediction models. Most methods are parametric and need manual tuning. We use machine learning based regression models to obtain a generic data prediction framework.

2.6 User Documentation

The web application will include a section dedicated to step by step tutorial for using various functions. Web pages containing information regarding utilization of application and its limitations will be present.

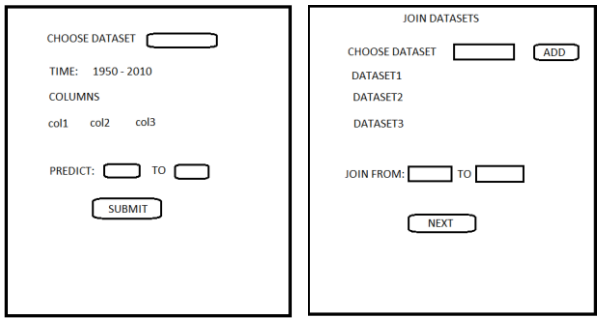
2.7 Assumptions and Dependencies

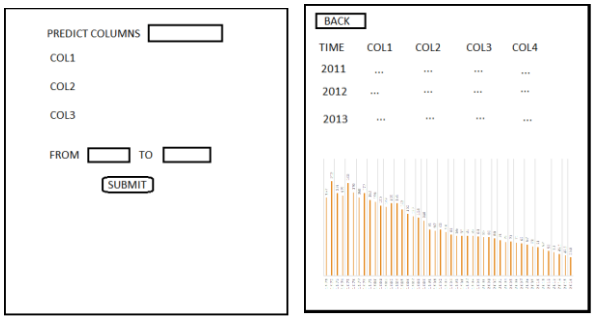
The application works based on existing OGD available on data.gov.in. It is assumed that the data provided is accurate and consistent with the actual facts. This is important since faulty or erroneous data may lead to wrong predictions.
Consequently, it is assumed that data obtained through the application will not be used for critical decisions.

3. External Interface Requirements

3.1 User Interfaces

The application will be accessed by users through a web interface. The website will contain several sections pertaining to various functionalities offered.





The backend components: Database, Prediction framework, web server will not have a user interface.

3.2 Hardware Interfaces

Any modern computer capable of running a modern web browser (preferably Google Chrome) with Internet connectivity would suffice for a user.
Any modern computer would be sufficient to meet the server requirements.

3.3 Software Interfaces

A Web interface is provided to the user for using the services. This would involve usage of HTML5, CSS3, Bootstrap and JavaScript.
Service layer would be developed in Python (version 2.7). Numpy, Sci-kit libraries would be used.
MySql database would be used for data storage.

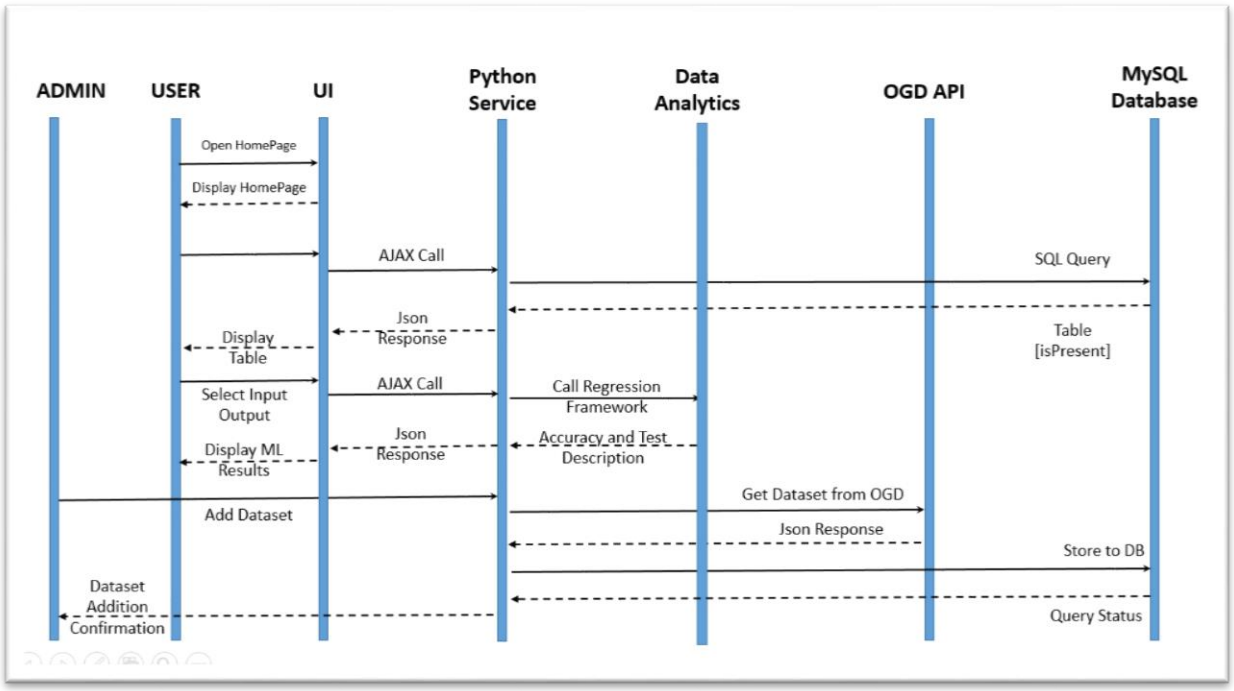


Figure 3 : Sequence Diagram

3.4 Communications Interfaces

A web browser will be required to access the application on the Internet. Communication between UI and Service layer would be facilitated using HTTP requests.

4. System Features

4.1 Choose Your Database

4.1.1 *Description and Priority*

Datasets will be organized by categories such as finance, agriculture, healthcare etc. Users may also search for the datasets by name. A User would be able to select database of his choice from those available at OGD. Many datasets have large amount of data. A user may select a subset of a dataset based on time and data columns.

This is a **High Priority** feature

4.1.2 *Stimulus/Response Sequences*

1. Users selects a category/subcategory: List of datasets available will be displayed.
2. User searches for dataset by name: Relevant databases are shown as list.
3. User selects a dataset: Dataset description is displayed: Short description of columns and number of data rows. A set of functions that can be performed using the dataset are displayed (eg. Visualization, Refine selection, prediction).
4. User selects a function: Appropriate interface for the selected function is displayed. ‘Refine selection’ will allow user to select subset of data based on time, location, data columns etc.

4.1.3 *Functional Requirements*

User should be able to select a dataset of his choice and see the relevant information (described above) after the selection. Necessary requirements to use this feature are:

1. Database should be accessible.
2. Web browser must be functional with Internet connectivity.

In case of database access failure, user should be shown a relevant error message.

4.2 Data Visualization

4.1.1 *Description and Priority*

Along with the dataset selected in ‘Choose Your Database’ phase, user would be shown several visualization options like pie charts, line graphs and tabular display of data. These will be rendered within the

web browser. An option to predict values will be shown in which user can select time period for prediction.

4.1.2 *Stimulus/Response Sequences*

User can choose to visualize columns of the dataset or predictions from our application. An option to

1. User selects a visualization option : The graphical visualization is displayed on the same page. User may select another option and view it without reloading.
2. User selects data prediction option : User is shown date range and data columns to select for prediction and visualization is updated accordingly.

4.1.3 *Functional Requirements*

Functional requirements for this feature are:

1. Database should be accessible.
2. Library used for graphical visualization must function properly.

If database access or visualization library fails to work, user must be notified using error messages.

4.3 Data Prediction

4.1.1 *Description and Priority*

Using the existing dataset instances, we would use appropriate machine learning techniques to make predictions for future values.

The accuracy of the prediction will vary based on data type and amount. Machine learning techniques require large amount of data. This feature may not be available for all datasets.

This is a **High Priority Feature**

4.1.2 *Stimulus/Response Sequences*

1. User selects prediction option: User will be shown selection for columns of the dataset that can be predicted and time period for prediction.
2. User selects required columns and time period: Predicted data values are displayed and visualization (if any) is updated.

4.1.3 *Functional Requirements*

Functional requirements for this feature are:

1. Database should be accessible.
2. Prediction component of application should be functioning and accessible.
3. Correlation can only be shown for numeric valued attributes

If database access or prediction component fails to work, user must be notified using error messages.

If prediction cannot be generated with sufficient accuracy, user must be notified instead of showing erroneous data.

4.4 Computation History Storage

4.1.1 *Description and Priority*

A user can choose to save his data filled on the website for future reference with just a button click. This data can later be seen through the Load feature and also, the corresponding predictions can be loaded directly.

This is **medium** priority feature.

4.1.2 *Stimulus/Response Sequences*

1. User saves the form data: By clicking on "Save" after seeing the predictions on homepage, user's data filled on this page would get stored and added to his history.

2. User opens the History Page: By clicking on "History" option, user is taken to the history page which lists all of user's saved data listed in reverse chronological order along with a view functionality to quickly view the predictions for that data.

4.1.3 *Functional Requirements*

Functional requirements for this feature are:

1. Database should be accessible.
2. User must be signed in to see his history.
3. Prediction component of application should be functioning and accessible.

If database access or prediction component fails to work, user must be notified using error messages.

If prediction cannot be generated with sufficient accuracy, user must be notified instead of showing erroneous data.

4.5 Feature Correlation

4.1.1 *Description and Priority*

A user can view correlation among selected attributes of a dataset. The correlation would be shown as a coloured heat map along with legends and scale.

This is **medium** priority feature.

4.1.2 *Stimulus/Response Sequences*

1. User selects the column for finding correlation: User has to fill a form selecting the attributes for which he wants to see correlation.

2. User submits column selection: A properly labelled coloured heat map would be shown which depicts the attribute correlation.

4.1.3 *Functional Requirements*

Functional requirements for this feature are:

1. Database should be accessible.

2. Backend python service should be up.
3. Correlation component should be working.

If database access or prediction component fails to work, user must be notified using error messages.

5. Other Non-functional Requirements

5.1 Performance Requirements

The web interface should be responsive and fast. In case of ongoing background operations, loading image should be displayed to indicate it. Visualization of data/predictions should be displayed quickly. Model used for prediction should predict outputs quickly (< 1 second) even for large datasets. Fetching necessary data from database and getting output from prediction model should be quick (few seconds at most).

5.2 Safety Requirements

Predictions made using the software may not be very accurate in some cases. The predictions should not be treated as truth. Any inferences obtained from the model should be verified from other sources for critical systems (such as healthcare).

5.3 Security Requirements

The website is publicly accessible and its function can be used by anyone. User may or may not identify himself/herself while using website's functions.

Database system and prediction model must have access restrictions. Addition or modification of data should only be possible for administrators.

Only publicly available data should be displayed on the website. Any confidential or sensitive data should not be available through the website.

5.4 Software Quality Attributes

Adaptability: The prediction and visualization models must adapt to new data added to database.

Performance: The web interface must be responsive and complete operations quickly(less than a second). Visualization and predictions should be quickly processed and displayed within a few seconds.

Availability: The website must be able to handle large number of user requests. Website downtime must be as low as possible.

Correctness: Data displayed on website should be correct (dependent on data source). Predictions should specify some correctness measure to help users decide accordingly.

Reusability: The same web interface and backend should be used for new data and accommodate new prediction models for improved accuracy.

Usability: Web interface should be clean and easy to use. Data predictions and visualization should be graphically displayed for easy interpretation. Selection of data/fields for a given dataset should be easy.

5.5 Business Rules

Database administrator should be able to add/modify data in database. Read access is given publicly. Webmaster should be responsible for changes/improvements to web interface. Modifications in prediction models must only be done by Data Scientist.
Users should be able to access the system through web interface only and shouldn’t make changes to underlying system.