ReactJS Project

Eazy-Admin by Sanele Manyela & Samkelwe Ngalimane

Executive Summary

The React application project aims to make easy activities associated with viewing and editing supplier and contract data. The application uses an API to read and store the data used.

System investigation

- **Software Purpose**

  The software application is being designed to view supplier and contract data as required and be able to access API data. The application will also allow for the contract and supplier data to be updated.

  The application has the following objectives:

  - Aid an administrative user in viewing data of suppliers and that of contractual agreements.
  - Simplify manipulation of data used by the application.

- **Software system capabilities**

  The software application will access API data and display supplier data and contract data as required. The application will also be able to update both the supplier and the contract data.

- **Software system actors**

  The application will be used by administration personnel.

- **Feasibility Study**
    - **Statement of the problem**

      Design and build a platform to view Supplier and Contract Data, using React for the frontend. The platform should be able to access the API Data and display it as required.

        - **Summary of findings**
            - **Technical feasibility**

              React (also known as **React.js** or **ReactJS**) is a free and open-source front-end JavaScript for building user-interface based on UI components. React will be used for creating the platform. React allows working with and making API calls.

            - **Legal feasibility**

              The application will be subject to Protection of Personal Information Act (POPIA) to protect supplier data and contract data. POIPA ensures customer's personal details are treated in a legitimate and respectful manner. The Protection of Personal Information Bill states that personal information must be used explicitly defined and lawful purposes related to a function or activity of your business. It is the business' responsibility to ensure a customer's information is complete, accurate, truthful, and up to date.

- **Project Planning**
    - **Project background**

      Description: This project is an endeavour to design and develop a platform using React as a choice of technology for the frontend development. The platform will be able to access API data and display it as required. The users of the final deliverable will be administrators who will view supplier and contract data.

Project start date: 07 December 2021

Project end date: 21 January 2022

Scheduling strategy: Schedule from project start date

Calendar: The project will use the standard Gregorian calendar. The project working days are Monday to Friday.

- **Project Team**

  Team members

  - Sanele Manyela
  - Samkelwe Ngalimane

  Roles

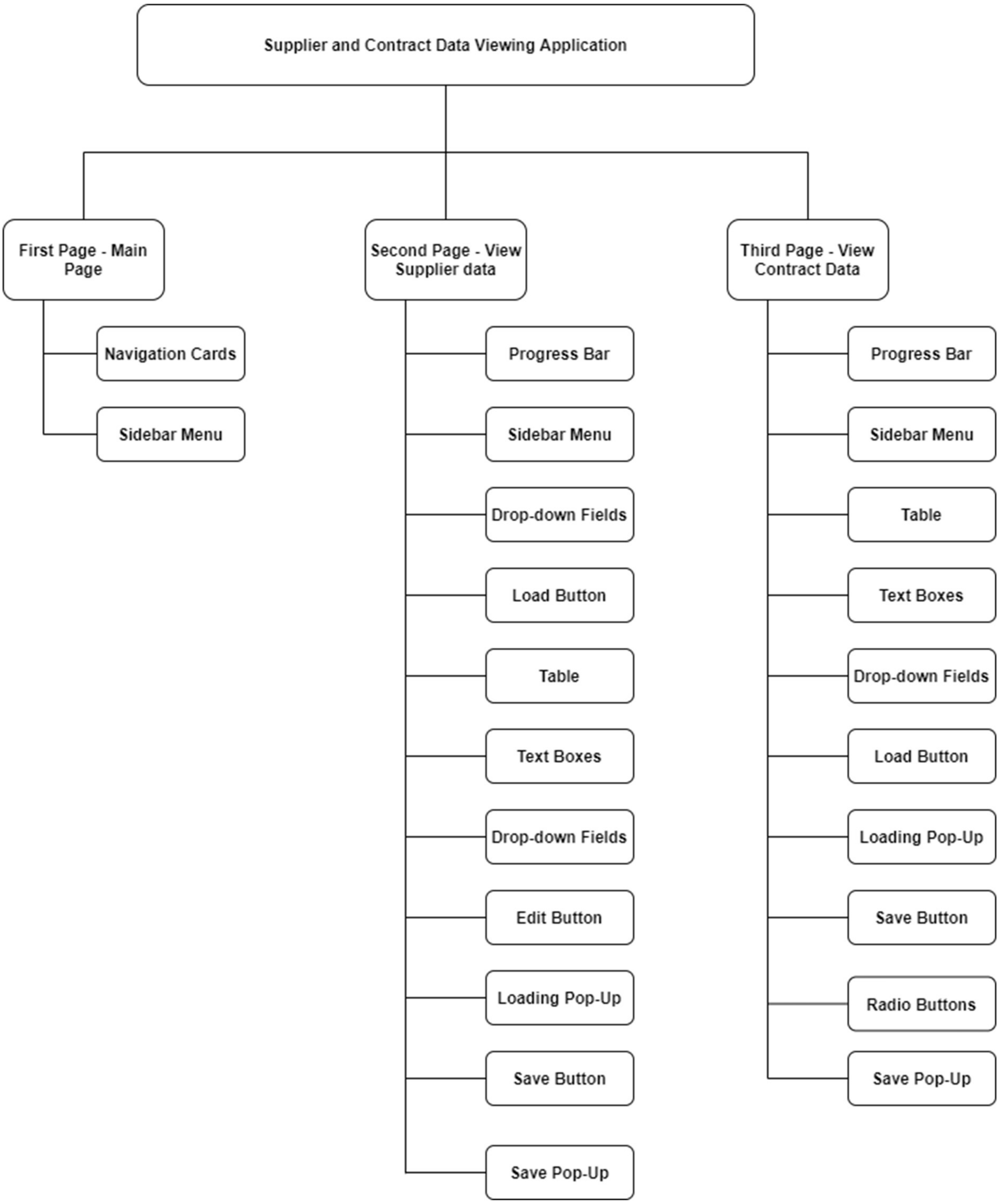
  The team roles for this project are dynamic in their nature, both team members are free to take on the roles of project manager, systems/software analysts, software developer, software tester, and software architecture.

- **Project Scope**

  The scope of the project specifies a requirement to design and build a platform to view supplier and contract data, using React for the frontend. The platform must be able to access API data and display it as required.

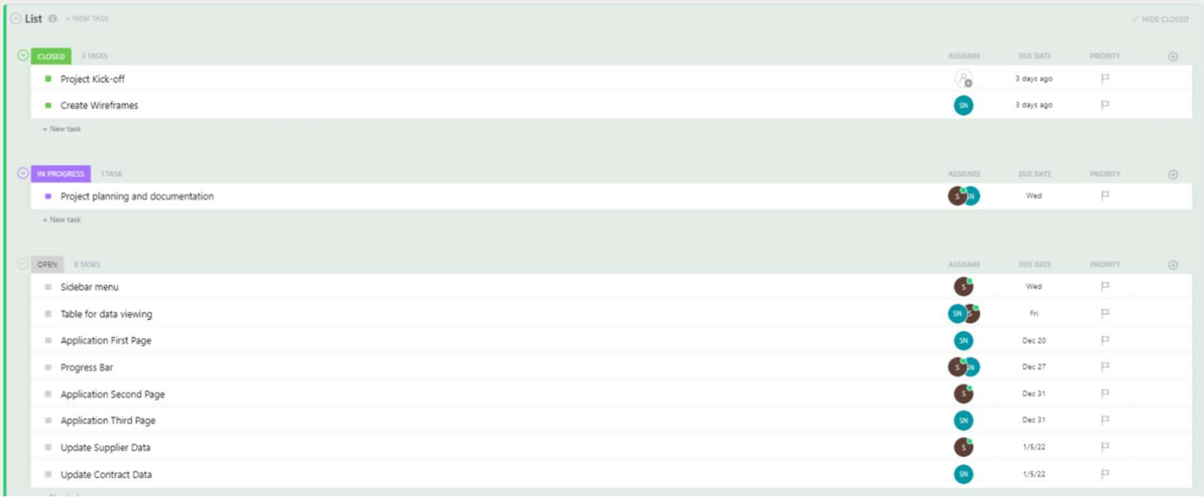
  The platform must be comprised of three pages: the front and main page with navigation to other two pages, the second page will display supplier data that can also be updated, and the third page will display contract data that can also be updated.

- **Product Breakdown Structure**



The illustrated figure above depicts the project’s product breakdown structure. The project deliverable will consist of five main components: the landing page component, the page component for viewing supplier data, the page component for viewing contracts data, the page component for updating supplier data, and the page component for updating contract data. Each of these components is further divided to self-contained components as shown in the illustration.

- **Assignment of Activities**



List + NEW TASK ✓ HIDE CLOSED

CLOSED 2 TASKS	ASSIGNEE	DUE DATE	PRIORITY
Project Kick-off		3 days ago	⚐
Create Wireframes	SN	3 days ago	⚐

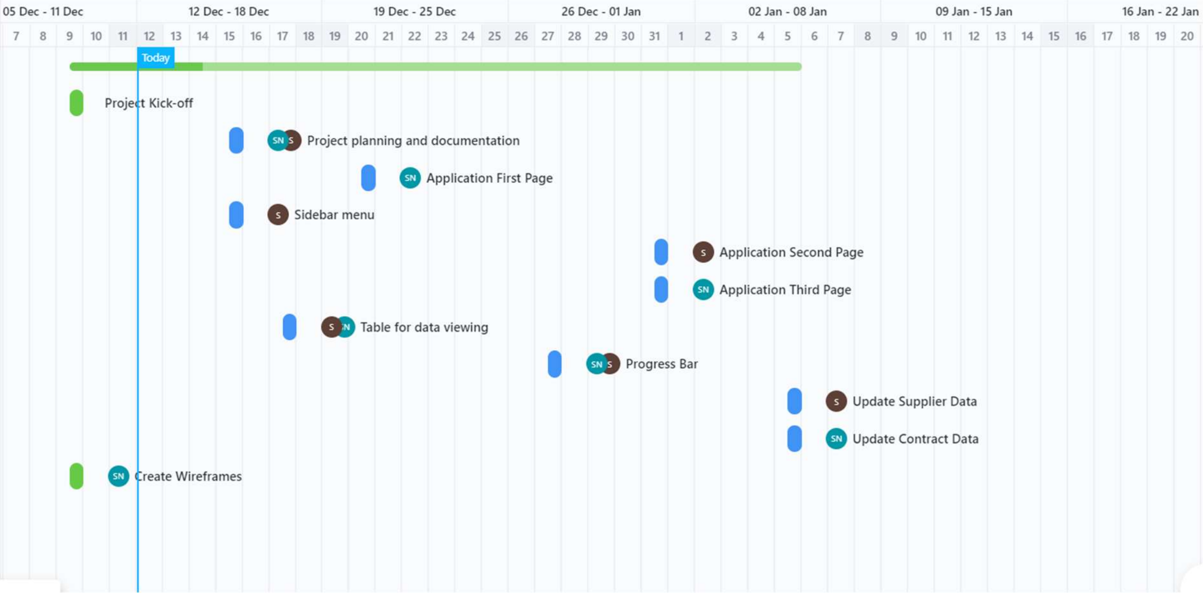
+ New task

IN PROGRESS 1 TASK	ASSIGNEE	DUE DATE	PRIORITY
Project planning and documentation	S SN	Wed	⚐

+ New task

OPEN 8 TASKS	ASSIGNEE	DUE DATE	PRIORITY
Sidebar menu	S	Wed	⚐
Table for data viewing	SN S	Fri	⚐
Application First Page	SN	Dec 20	⚐
Progress Bar	S SN	Dec 27	⚐
Application Second Page	S	Dec 31	⚐
Application Third Page	SN	Dec 31	⚐
Update Supplier Data	S	1/5/22	⚐
Update Contract Data	SN	1/5/22	⚐

- **Timeline**



- **Software System Analysis**
  - **Information and requirements gathering**

    Information and requirements were gathered using group meeting techniques and the scope document.
    - **Definition of requirements**
      - The application should contain three pages
        - Front Page, with cards
        - Second Page: With Supplier Data
        - Third Page with Contract Data
      - When each page loads, a progress bar should appear, notifying the user the data is loading.
      - Front Page Tile Menu should have three cards.
        - There should be a card that navigates to each of the following:
          - Navigate to the following URL: www.google.com
          - Second Page: With Supplier Data
          - Third Page with Contract Data
      - The application requires a sidebar menu, with drawer functionality, the sidebar menu should have a selection for Home, Supplier Reporting, Contract Reporting.
      - The second page should have two dropdown fields and a button, as well as a table, which should be filtered by the dropdown selection.
        - Drop Down 1- Year
        - Drop Down 2 – Month
        - Button – Load
        - Table – Data from the Supplier Master Data API.

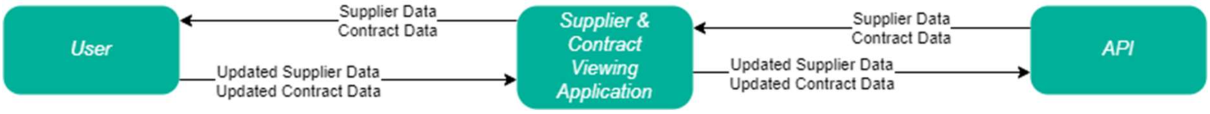
        - OnClick of Table Row, Supplier Captured Data should be displayed in Editable Field (Either Drop Down or Textbox) and a button to Save.

- The Save button should be greyed out and only useable when data is changed in the fields
- A loading popup should be displayed
- A Saved popup should be displayed

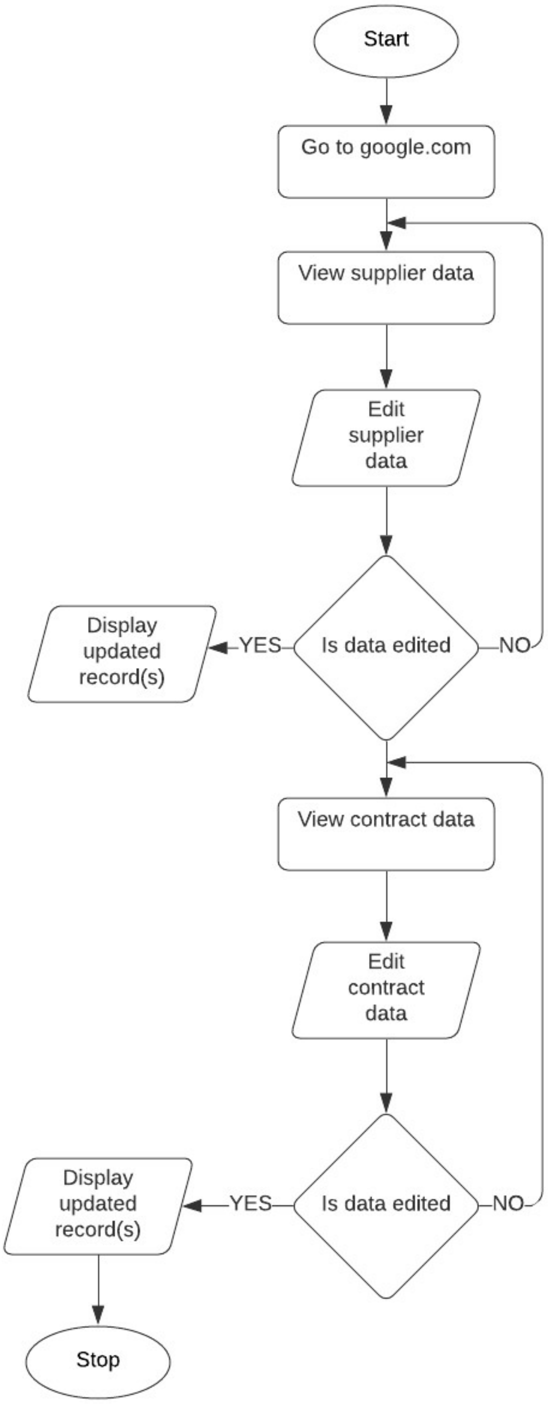
- Third Page should have only a table, which should load when the page is selected.
  - Table – Data from the Contract Master Data API.
  - OnClick of Table Row, Contract Capture Data should be displayed in Editable Field (Either Drop Down or Textbox) and a button to Save.
    - The Save button should be greyed out and only useable when data is changed in the fields
    - A loading popup should be displayed
    - A Saved popup should be displayed
    - The Following fields should be drop down menu
      - LEDSplit
        - MBSA
        - DTBSA
        - MBFS
        - Trucks
        - Vans
        - SMH
      - Classification
        - A
        - B
    - The following fields are radio buttons
      - Plant Relevant
        - Yes
        - No
      - Saved to Server
        - Yes

        - No
    - All date fields should display today's date
- **Prioritisation of Requirements**
    1. Sidebar menu
    2. Table for data viewing
    3. Application First Page
    4. Progress Bar
    5. Application Second Page
    6. Application Third Page
    7. Update Supplier Data
    8. Update Contract Data

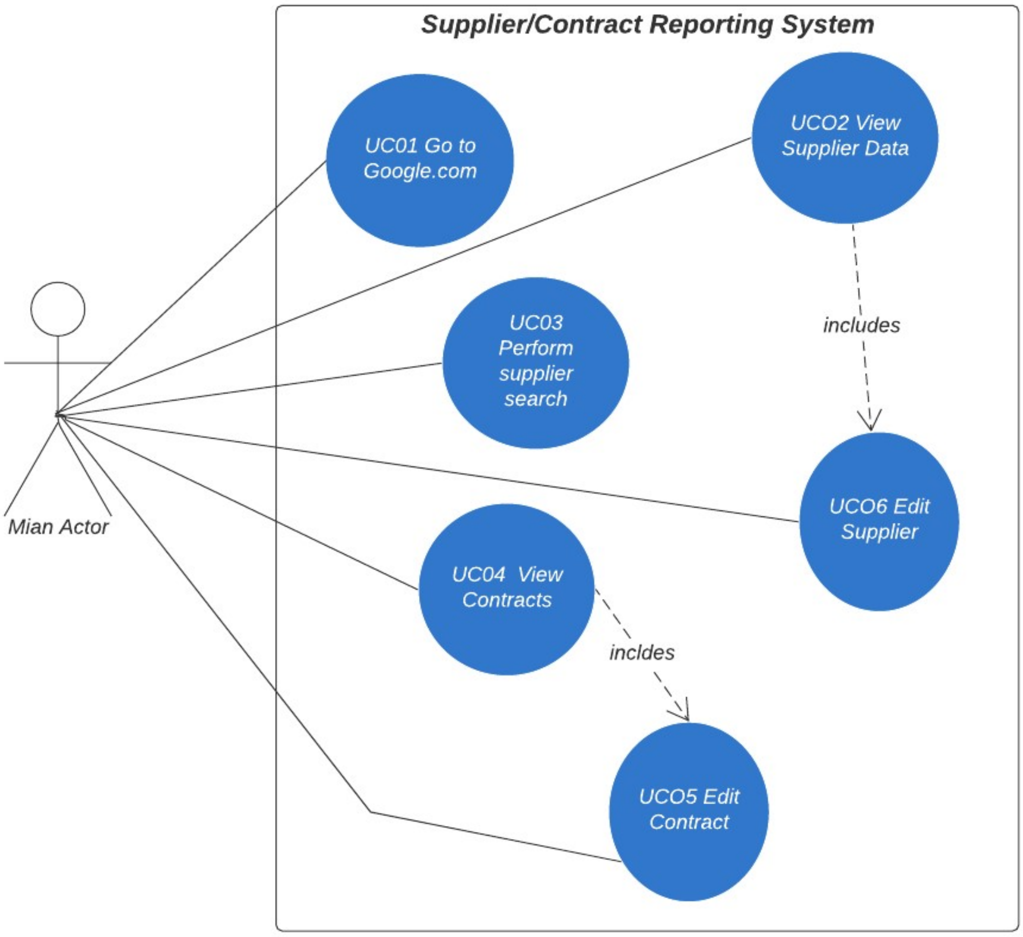
- **Software System Design**
    - **Context Diagram**



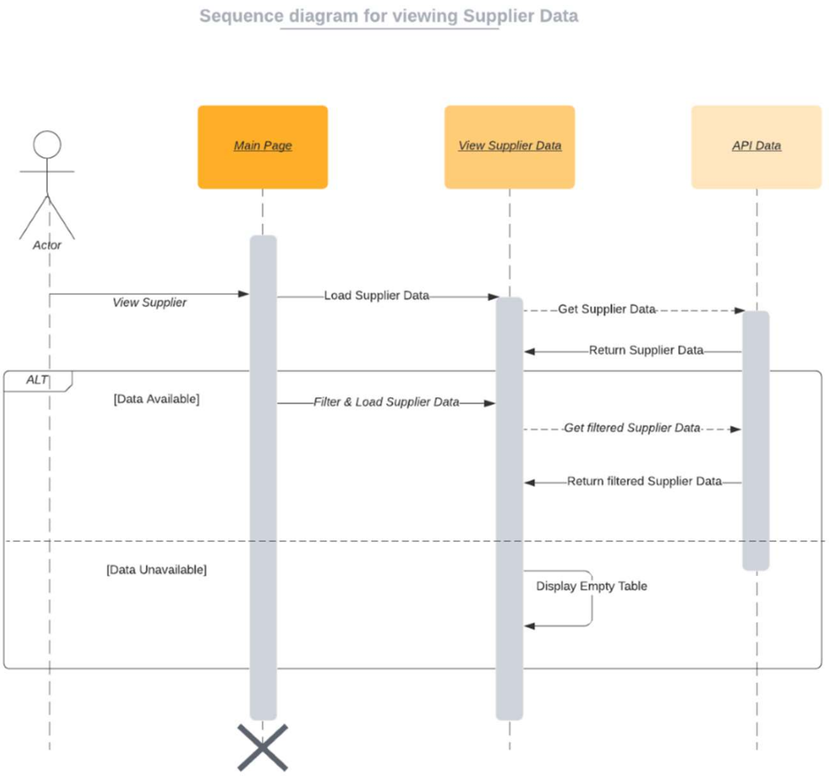
- ### Flowchart Diagram



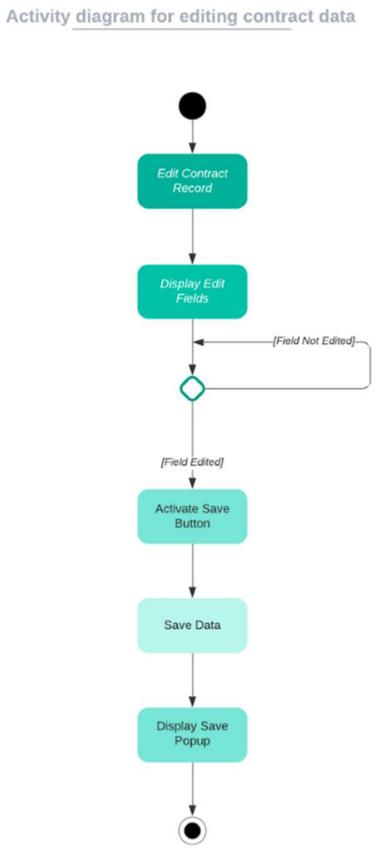
- **Use Case Diagram**



- ### Sequence Diagram

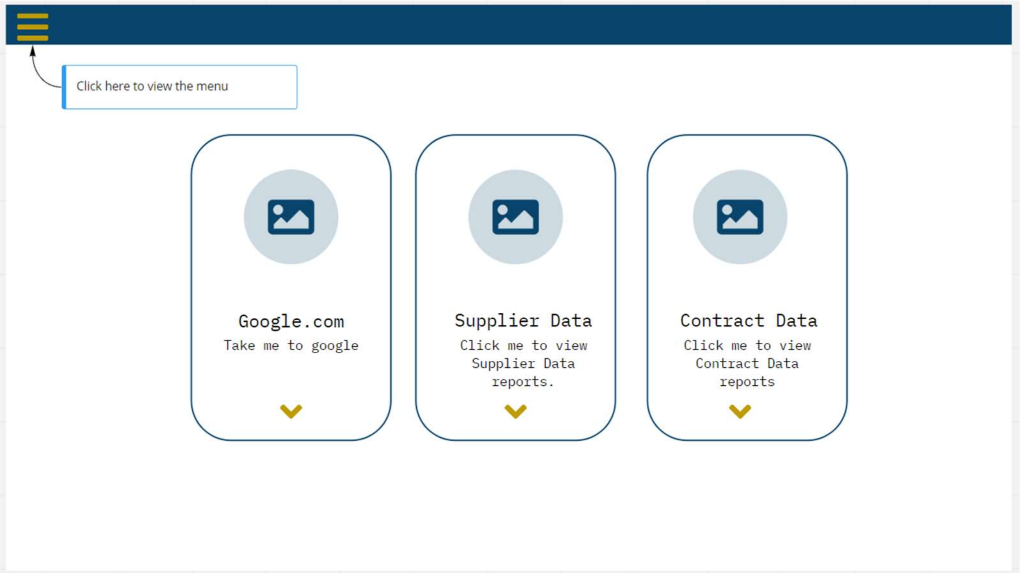


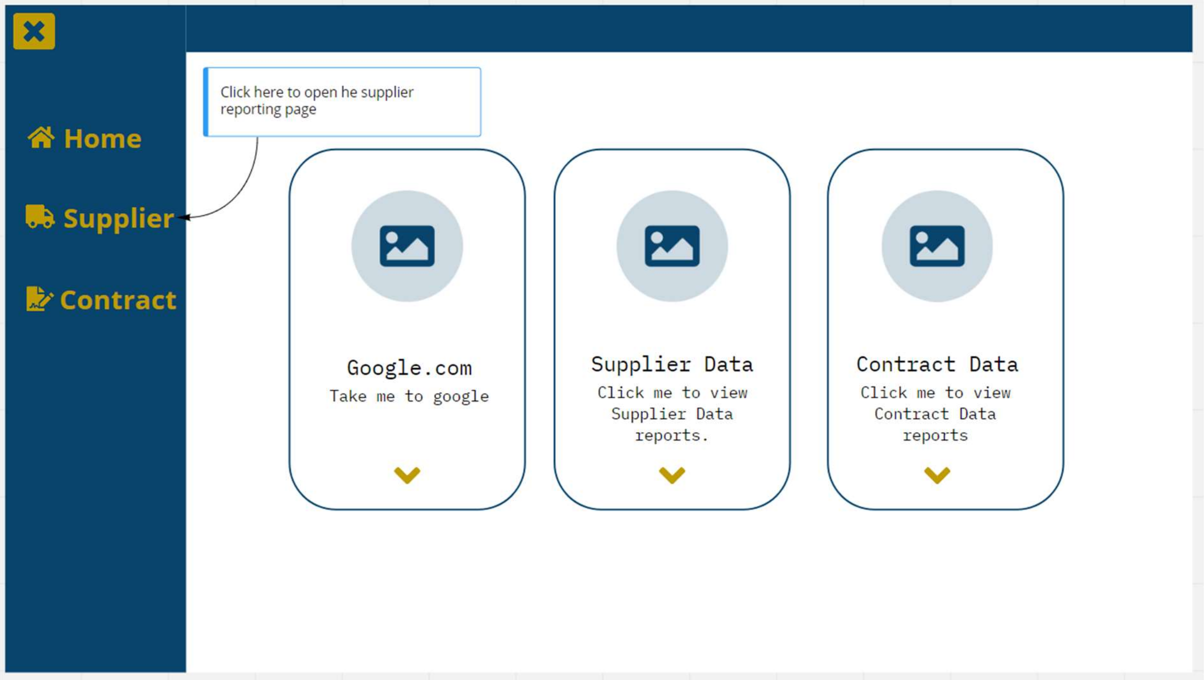
- ### Activity Diagram



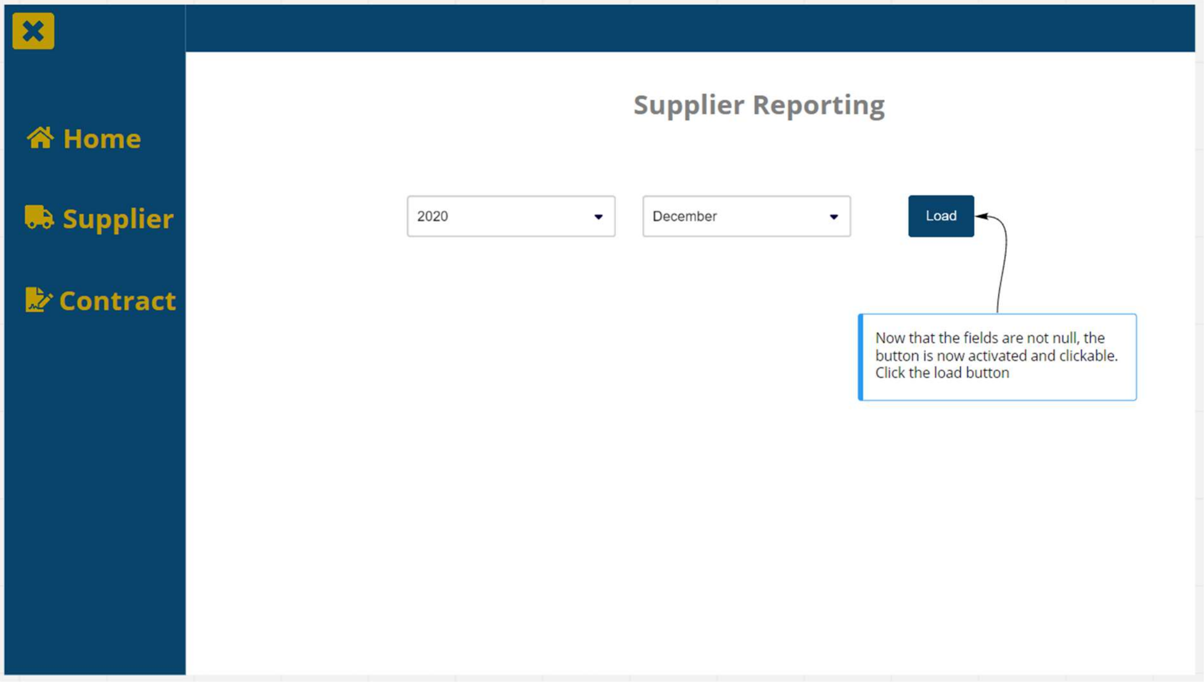
- ### User Interface Design and consideration

Home Page


Click here to view the menu
Google.com
Take me to google
Supplier Data
Click me to view Supplier Data reports.
Contract Data
Click me to view Contract Data reports


Home
Supplier
Contract
Click here to open he supplier reporting page
Google.com
Take me to google
Supplier Data
Click me to view Supplier Data reports.
Contract Data
Click me to view Contract Data reports

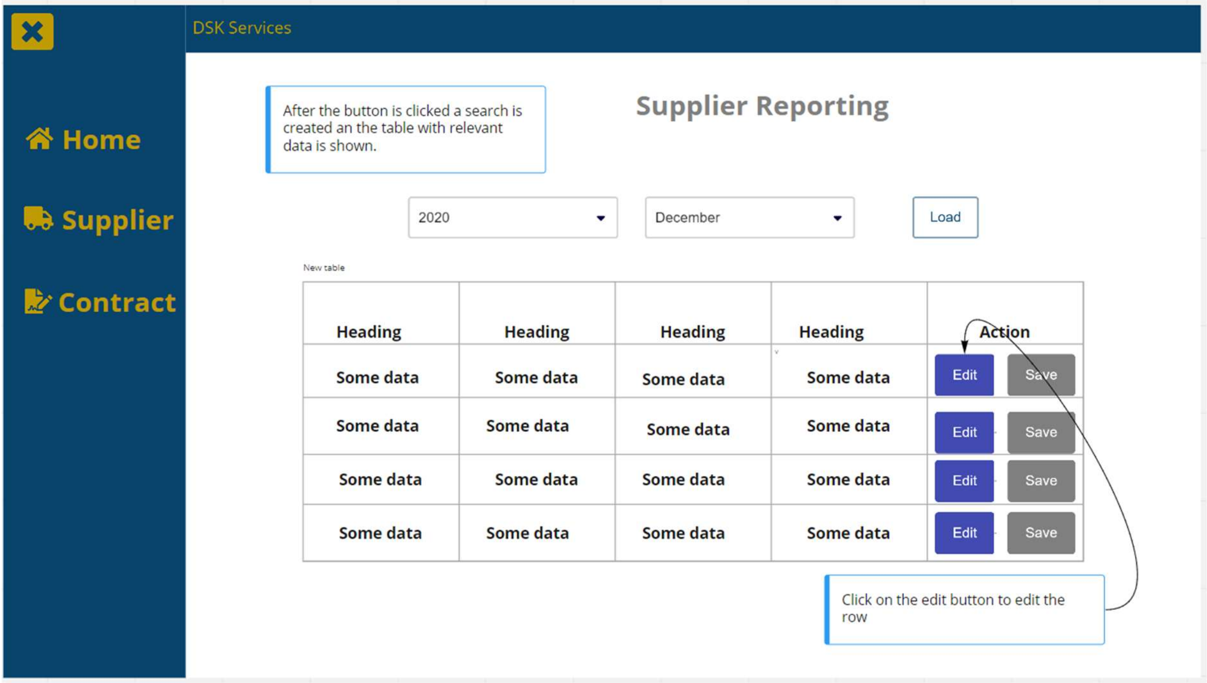
Supplier Page



Supplier Reporting

2020 | December

Now that the fields are not null, the button is now activated and clickable. Click the load button



After the button is clicked a search is created an the table with relevant data is shown.

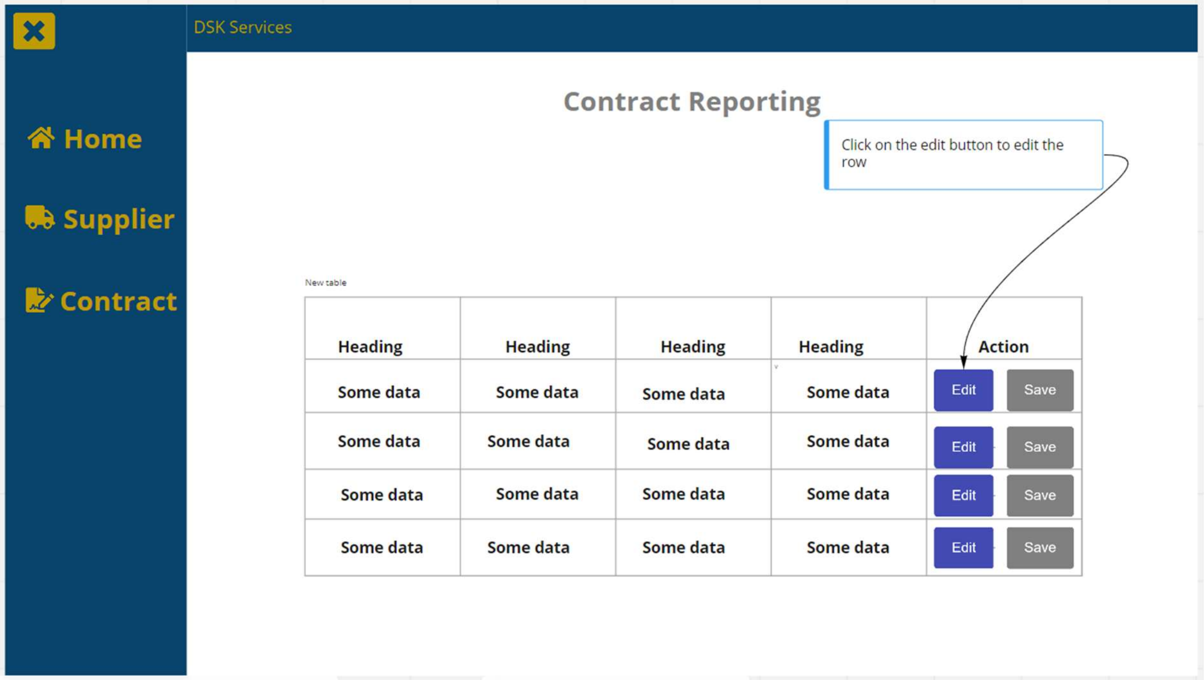
Supplier Reporting

2020 | December

Heading	Heading	Heading	Heading	Action
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save

Click on the edit button to edit the row

Contract Page

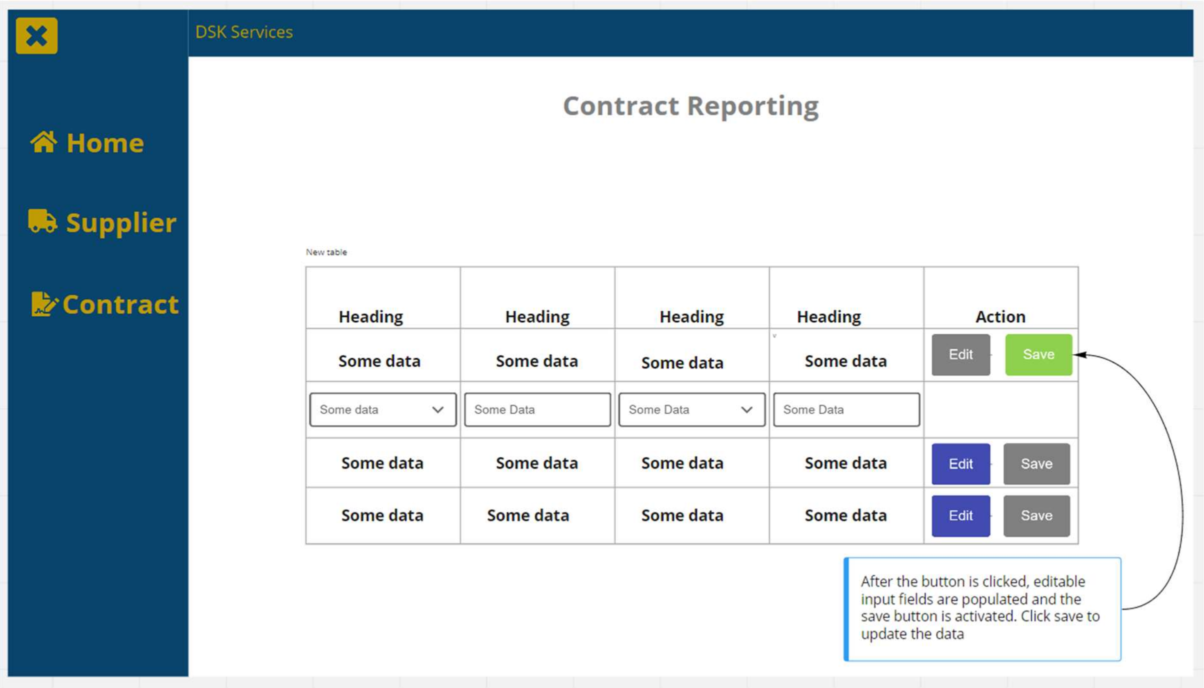


Contract Reporting

Click on the edit button to edit the row

New table

Heading	Heading	Heading	Heading	Action
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save



Contract Reporting

New table

Heading	Heading	Heading	Heading	Action
Some data	Some data	Some data	Some data	Edit Save
Some data	Some Data	Some Data	Some Data	
Some data	Some data	Some data	Some data	Edit Save
Some data	Some data	Some data	Some data	Edit Save

After the button is clicked, editable input fields are populated and the save button is activated. Click save to update the data

- ## Implementation

  Clear outline of the development technologies that will be used

  - **Test Strategy – Development Driven Testing**
    - First Page - Home Tests

      1. Test Case 1: [
         IDENTIFIER: Sliding bar navigation
         TEST CASE: Test if sliding bar menu item can navigate to the Supplier Page.
         PRECONDITIONS: None
         INPUT VALUES: None
         EXECUTION STEPS: On the application landing page, click the three-line icon at the left top corner of the interface, on display of a slide bar menu select Supplier.
         OUTPUT VALUES: N/A
         POSTCONDITION: The application navigates to the Supplier interface.
         ]

      2. Test Case 2: [
         IDENTIFIER: Sliding bar navigation
         TEST CASE: Test if sliding bar menu item can navigate to the Contract Page.
         PRECONDITIONS: None
         INPUT VALUES: None
         EXECUTION STEPS: On the application landing page, click the three-line icon at the left top corner of the interface, on display of a slide bar menu select Contract.
         OUTPUT VALUES: N/A
         POSTCONDITION: The application navigates to the Contract interface.
         ]

3. Test Case 3: [
   IDENTIFIER: Sliding bar navigation
   TEST CASE: Test if sliding bar menu item can navigate to the Home Page.
   PRECONDITIONS: The application must be in the Contract Page interface.
   INPUT VALUES: None
   EXECUTION STEPS: On the application landing page, click the three-line icon at the left top corner of the interface, on display of a slide bar menu select Contract. When inside the Contract Page interface, click in the slide bar again and navigate to Home.
   OUTPUT VALUES: N/A
   POSTCONDITION: The application navigates to the Home interface.
   ]

4. Test Case 4: [
   IDENTIFIER: Navigation with Cards
   TEST CASE: Test if the Google navigation card navigates to Google.com
   PRECONDITIONS: None
   INPUT VALUES: None
   EXECUTION STEPS: On the application landing page, click the Google.com card, the first card from the left of the three navigation cards.
   OUTPUT VALUES: N/A
   POSTCONDITION: The application redirects to a Google.com search interface.
   ]

5. Test Case 5: [
   IDENTIFIER: Navigation with Cards
   TEST CASE: Test if the Supplier navigation card navigates to Supplier interface.
   PRECONDITIONS: None
   INPUT VALUES: None
   EXECUTION STEPS: On the application landing page, click the Supplier card, the second and centre card from the left of the three navigation cards.
   OUTPUT VALUES: N/A
   POSTCONDITION: The application redirects to Supplier Page interface.
   ]
6. Test Case 6: [
   IDENTIFIER: Navigation with Cards
   TEST CASE: Click the button in the Contract card to navigate to the application’s Contract Page
   PRECONDITIONS: None
   INPUT VALUES: None
   EXECUTION STEPS: On the application landing page, click the Contract card, the third and right-most card from the left of the three navigation cards.
   OUTPUT VALUES: N/A
   POSTCONDITION: The application redirects to Contract Page interface.
   ]

- Third Page Tests – Contract

1. Test Case 1: [

   IDENTIFIER: Contract Data Retrieval

   TEST CASE: Test if the Application retrieves the data that will be displayed on the Contract Page table

   PRECONDITIONS: Implement a standard method of writing to the console.

   INPUT VALUES: None

   EXECUTION STEPS: On the application landing page, click the Contract card.

   OUTPUT VALUES: N/A

   POSTCONDITION: The application navigates to the Contract Page interface and the data that is retrieved from the Contract Master Data API is displayed

   ]

2. Test Case 2: [

   IDENTIFIER: Display Contract Data on table with some fields omitted.

   TEST CASE: Test if the application can navigate to the Contract Page and view a table displaying data.

   PRECONDITIONS: None

   INPUT VALUES: None

   EXECUTION STEPS: On the application landing page, click the Contract card at the centre of the interface.

   OUTPUT VALUES: N/A

   POSTCONDITION: The application navigates to the Contract interface and displays a table with contract master data.

   ]

3. Test Case 3: [

IDENTIFIER: Contract Page table capability to handle onClick events.
TEST CASE: Test if the Contract Page table is capable of handling onClick events..
PRECONDITIONS: Implement a standard method of writing to the console.
INPUT VALUES: None
EXECUTION STEPS: On the application landing page, click the Contract card and Navigate to the Contract Page. inside the Contract Page interface, click rows at random.
OUTPUT VALUES: N/A
POSTCONDITION: The application navigates to Home interface.
]

- Second Page Tests - Supplier

- **System Tests**

- **Documentation and description of each code artefact/component**

- ## User Manual

  - **Description of the software system.**

The software application is designed to view supplier data by passing date and contract data as required and be able to access API data. The application will also allow for the contract and supplier data to be updated.

The application is capable of:

- Smooth and easy navigation.

- Navigates to Google.com, navigates to Supplier Data, navigates to Contract Data.
- Supplier Data – displays a Supplier data table depending on the selected month and year.

The table has clickable rows that bring up a modal filled with values related to each individual row. The values in the modal may be updated.

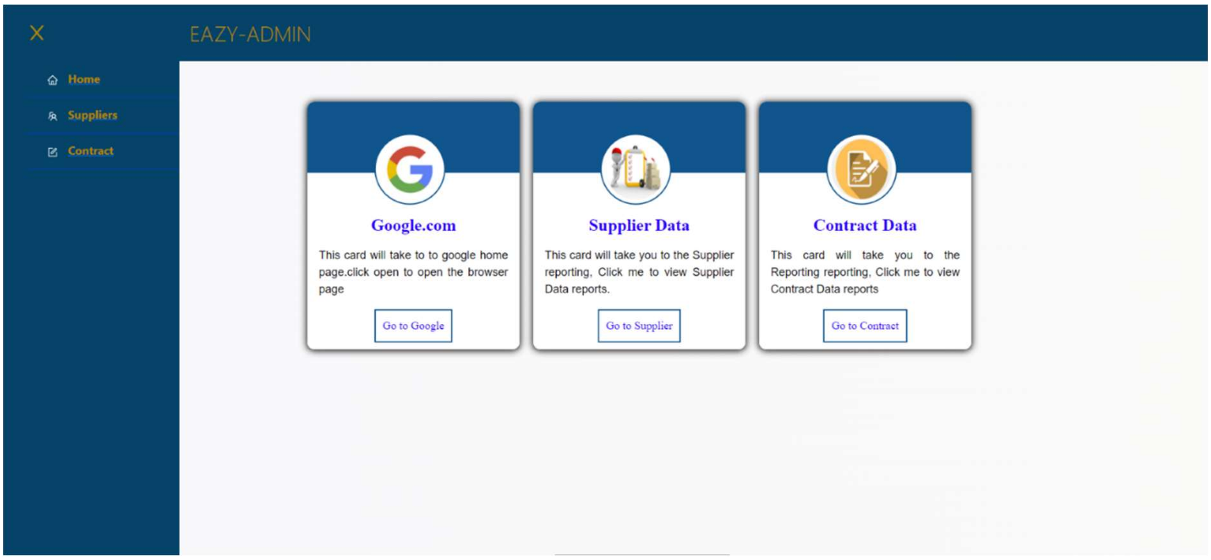
- Contract Data – displays a Contract data table.

The table has clickable rows that bring up a modal filled with values related to each individual row. The values in the modal may be updated.

- **Environment to run the system in.**
- **Technologies used by the system and requirements.**
- **Application inputs.**

- **User Interface**

1. **Application Landing Page**



Google.com

This card will take to to google home page.click open to open the browser page

Supplier Data

This card will take you to the Supplier reporting, Click me to view Supplier Data reports.

Contract Data

This card will take you to the Reporting reporting, Click me to view Contract Data reports