SRI SAIRAM ENGINEERING COLLEGE

DYNAMIC UI FRAMEWORK

TEAM : TRAIL BLAZERS

GITHUB URL : https://github.com/Deepak-n-468/duif-cafe

S No.	College Id	Name	Pod Position	Mail id
1.	ET01CB01	DR. M. ANANTHI	Project Manager	hod.csbs@sairam.edu.in
2.	ET10CJ01	DR. M. NITHYA	Project Mentor	hod.cj@sairam.edu.in
3.	SEC20CB016	DEEPAK N	Member	sec20cb016@sairamtap.edu.in
4.	SEC20CB017	PRAVEEN BALAJI S	Member	sec20cb017@sairamtap.edu.in
5.	SEC20CB046	KRISHNA KUMAR B	Member	sec20cb046@sairamtap.edu.in
6.	SEC21CB003	THARUN RAJ R G	Member	sec21cb003@sairamtap.edu.in
7.	SEC21CB004	RAHUL V	Member	sec21cb004@sairamtap.edu.in
8.	SEC21CB033	JAYASHREE T	Member	sec21cb033@sairamtap.edu.in
9.	SEC21CB037	SHYAM V	Member	sec21cb037@sairamtap.edu.in
10.	SEC21CJ028	KARTHIKEYAN A	Member	sec21cj028@sairamtap.edu.in
11.	SEC21CJ037	MUTHU AANAND.S U	Member	sec21cj037@sairamtap.edu.in
12.	SEC21CJ057	KAMALESH KUMAR SARAVANAN	Member	sec21cj057@sairamtap.edu.in
13.	SEC21CJ038	RUTHI SHANKARI S	Member	sec21cj038@sairamtap.edu.in
14.	SEC21CJ014	YOGESH M	Member	sec21cj014@sairamtap.edu.in

Technical Solution Approach

1.Introduction

1.1. About this document

It specifies the approach of implementing the entire workflow of a Pluggable component in a Dynamic UI Framework.

1.1.1. Purpose & Scope of the document

- The objective is to collect and analyze all assorted ideas that have come up to define the Pluggable Components in a Dynamic UI Framework.
- It provides a detailed overview about the rendering of those feature based pluggable components that are mutable(by database entries).
- It explains how the features of a component through server configuration can be changed.
- Specifies the functionality and reusability of a pluggable component.

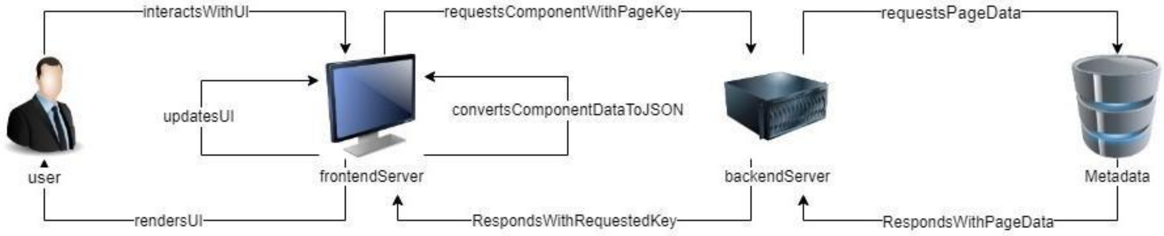
2.Component Design

2.1. Component Design Diagram

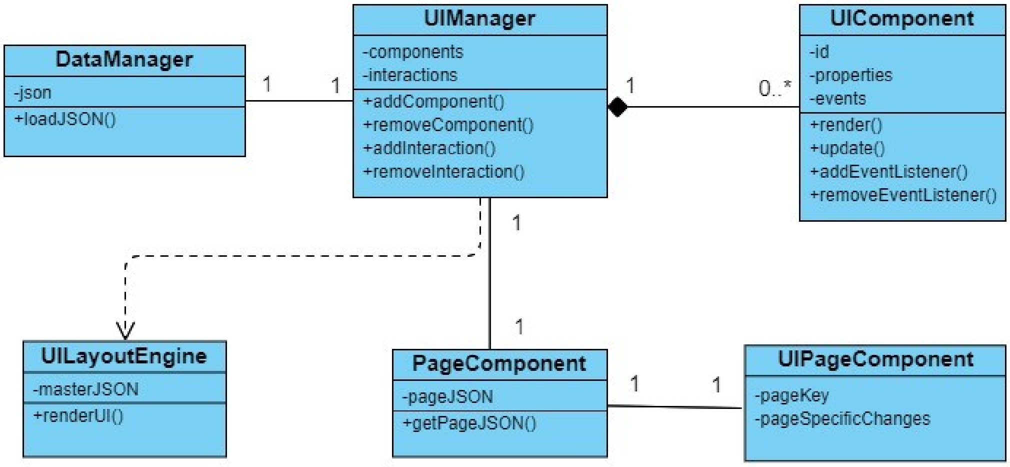
To enable interactivity on a web page, metadata is stored in a database. This metadata is used to retrieve or fetch data and build the feature-based components. Once the components have been built, the page key is passed to render the component using React. React is a JavaScript library that simplifies the process of building user interfaces. It allows for efficient updates to be made to the web page without requiring a complete refresh. By rendering the component in React, users can experience seamless and interactive web pages, improving the overall user experience.

2.1.1. Overall Workflow

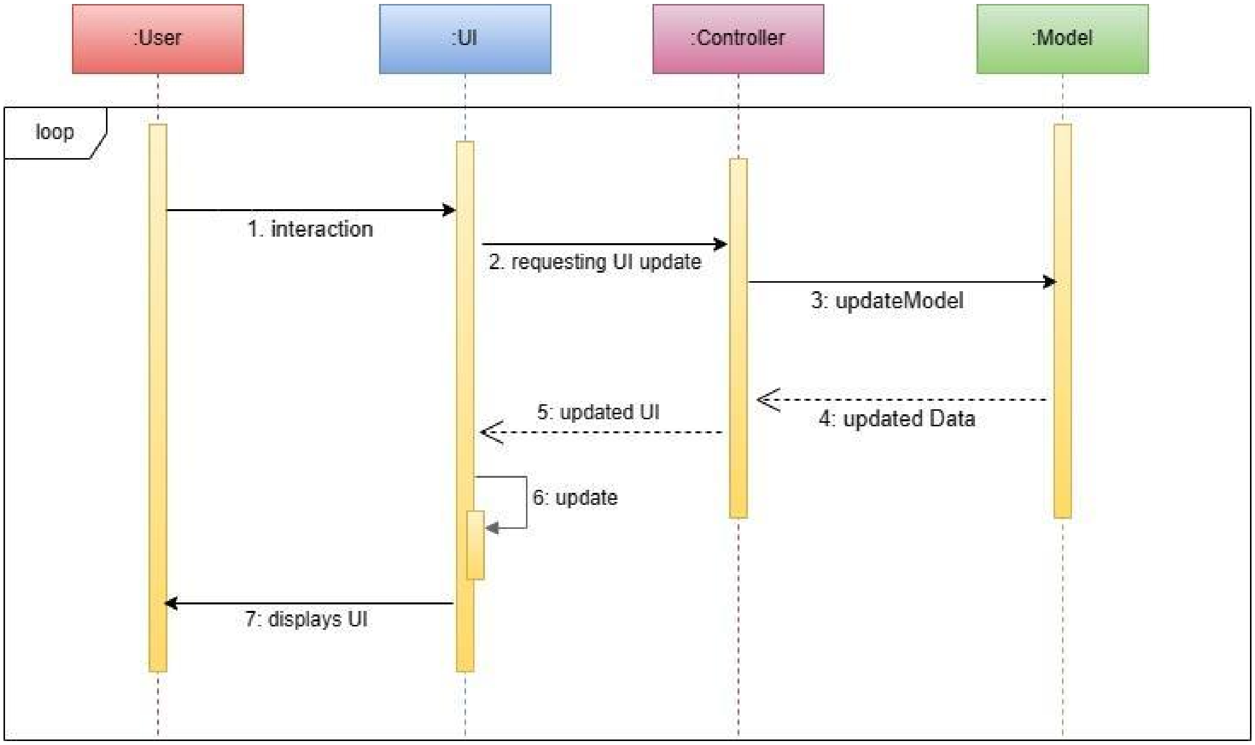
This section should explain the details of end to end workflow with the method level details. Below is the sample diagram just for reference.



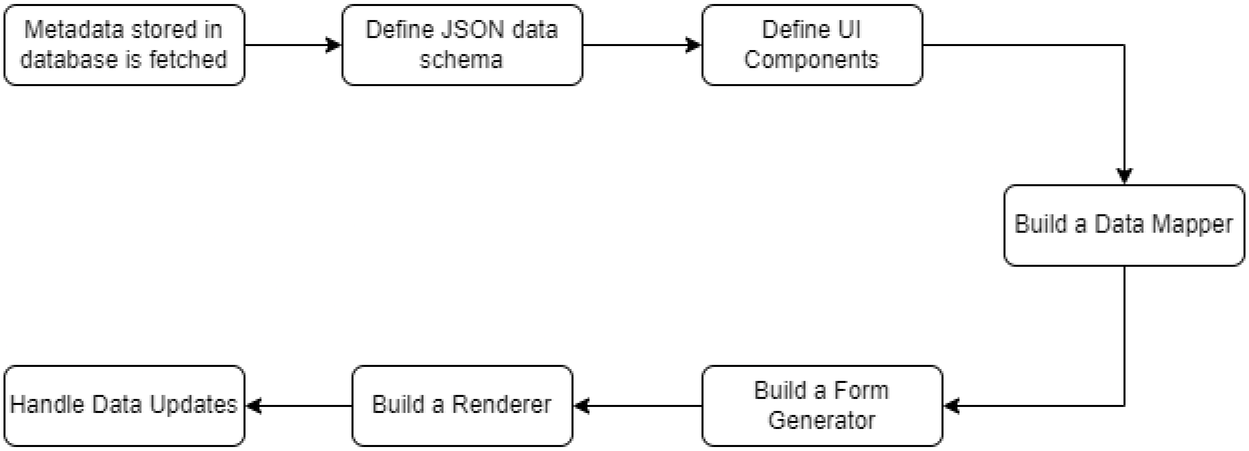
CLASS DIAGRAM :



SEQUENCE DIAGRAM:



2.1.2. Low level Design



- Define JSON data schema: we would first need to define a JSON data schema that describes the structure and properties of the data that will be used to render the UI. This could include things like the data types, field names, and validation rules for each property.

- Define UI components: Using React, we would define the UI components that will be used to render the JSON data. You could use tools like React-Bootstrap or Material UI to quickly build reusable UI components that can be easily customized.

- Build a Data Mapper: we would then build a data mapper that maps the JSON data to the UI components. The data mapper would take the JSON data and dynamically generate the corresponding UI components based on the schema you defined in step 1.

- Build a Form Generator: we would build a form generator that allows users to create and edit JSON data using the same dynamic UI components. The form generator would take the JSON schema and dynamically generate a form that allows users to create or edit the data in a user-friendly way.

- Build a Renderer: we would build a renderer that takes the JSON data and renders it as a read-only view using the same dynamic UI components. This would allow you to display the JSON data to users in a way that is easy to read and understand.

- Handle Data Updates: Finally, we would handle data updates in real-time by using state management tools like Redux or React Context. This would allow you to update the UI components in real-time as users interact with the data, providing a responsive and dynamic user experience.

3.Technology & Frameworks to be used

1. React Js.
2. Node Js.
3. SQL Database.

4.Solution Approach

1. The design diagram describes creating a Dynamic UI Framework.
2. Initially, the dynamic UI component will be fetching the data that are stored in the database.
3. The fetched data will be converted into JSON format.
4. The component will be utilizing the information stored in the format for the html tags and other required data.
5. Javascript files are loaded for event handlers to perform(Hydrate).