TMA Solutions

CANTEEN FOOD ORDERING APPLICATION

Truong Quoc Danh

Contents

TMA Solutions

Application Introduction

Objective:

Reduce waiting time when ordering food at the canteen.

Solution:

A mobile application that allows students to pre-order food and receive an invoice directly on their phones.

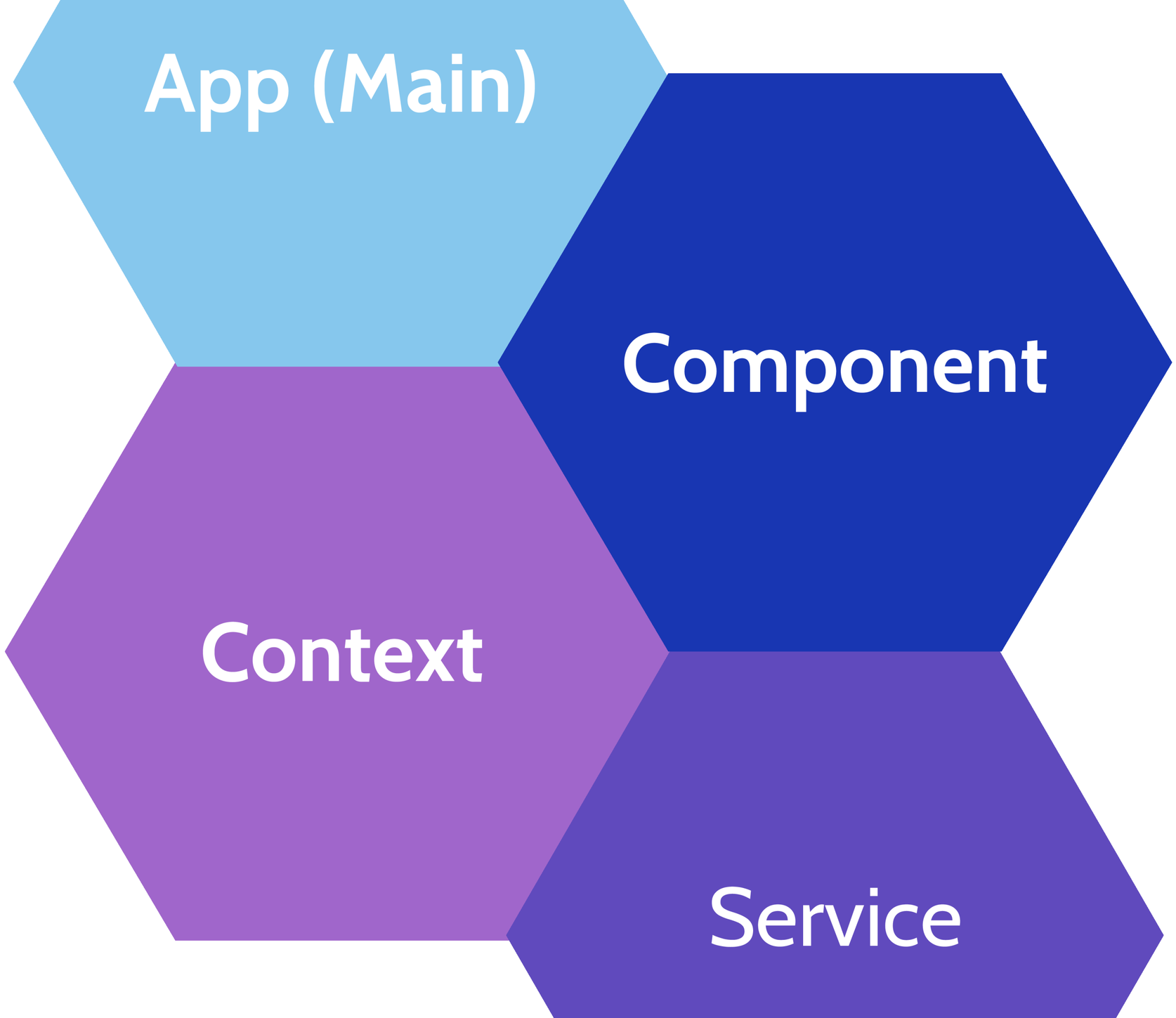
Benefits:

- Saves time
- Provides a convenient and quick experience
- Reduces overcrowding during peak hours

Using ReactJS for Front-End Design

The website interface for the food ordering application for the canteen consists of four main sections.

Sections of the Website



App Component

The customform component is a custom form designed to adjust the functionality for adding and editing items.

Header component purpose adjusts the header for App.js to custom.
SiderMenu to adjust Sidebar and manage state to be able to open and close the Sidebar

---

The Splash Screen component is designed to create an introductory page for the website.

Function Components

The FoodComponent helps the admin display a list of food items, utilizing useContext to manage the state of the list and retrieve data from the FoodService.

The AddFoodComponent allows the admin to add food items, using the Context API to manage state and share data via the FoodService.

The EditFoodComponent is a form similar to the AddFoodComponent, allowing for the editing of food item information by searching for it by ID.

The DeleteFoodComponent allows the admin to delete food items by ID by clicking the Delete button in the same row as the item in the list.

The context of the web consists of two parts:

AppContext & FoodContext

The AppContext is used to manage state and share userData across components such as Header, SiderMenu, and others.

The FoodContext is used to manage the state of components related to food functions.

Steps to Manage State in React JS:

The process for implementing state management in React can be described as follows:

1

Step 1:

Create component

2

Step 2:

Import Hook useState, useEffect from React

3

Step 3:

Initializing State in React

4

Step 4:

Using useEffect to Monitor State Changes in React

5

Step 5: (Optional)

Setting State with New Values in React When State Changes or Through Event Handlers

Steps to Use Context API in React JS

Process for Implementing State Management in React can be described as follows:

1

Step 1:

Create context

2

Step 2:

Import Hook useState, useEffect, createContext from React

3

Step 3:

In the context, instantiate a provider to provide value to the context

4

Step 4:

Wrap components that manage state into a provider

5

Step 5:

Use useContext in child components to access data

MOBILE APPLICATION

- FLUTTER
- GETX

Why?

Flutter

- Provides flexible widgets for UI design.
- One source code for both iOS and Android.
- Instant changes without rebooting.

GetX

- Easily track and update data
- Integrates many features, concise syntax, and easy to learn

The backend of the application

- Spring Boot
- SQL Server
- Data Mapping
- Spring Data JPA

Spring Boot & SQL Server

Accelerated Development: Quick and efficient coding due to clear structure.
Easy Maintenance: Layered architecture (Controller, Model, Repository) facilitates maintenance and upgrades.
Familiar Language: Java is widely used and easy to learn.

SQL Server: A powerful and flexible database management system widely used across various fields, enabling easy data validation and modification, along with table initialization through Java Spring Boot.

Data Mapping & Spring Data JPA

Direct Relationships: Establish direct relationships between data from multiple sources simultaneously.
Data Mapping: Leverages the robust features of Spring Data JPA for easier database interactions.

Seamless Integration: Compatible with Spring Framework, supporting Dependency Injection and EntityManager.
Custom Repository Support: Facilitates integration of custom repository code.
Auto Query Generation: Creates query methods from names, minimizing repetitive CRUD code.

DEMO

THANK YOU