# Technical Document: SmartEats Application

*Last updated: 23/05/2023*

## 1. Introduction

SmartEats is a mobile application designed to help users adopt healthy eating habits and maintain a regular workout routine. The application provides users with a collection of healthy recipes, workout routines, and a platform for connecting with other health-conscious users. Additionally, it allows users to take a photo of their food and receive information about the ingredients and nutrition.

This technical document serves as a guide for developers working on the SmartEats application. It outlines the technology stack, development practices, and project organization.

## 2. Technology Stack

The SmartEats application is built using the following technology stack:

- **TypeScript**: A statically-typed superset of JavaScript that enhances development productivity and code quality.
- **React Native**: A framework for building cross-platform mobile applications using JavaScript and React.
- **Firebase**: A cloud-based platform that provides backend services such as authentication, database, and storage.
- **Express.js**: A minimal and flexible web application framework for Node.js, used for building the backend server.
- **Jest**: A JavaScript testing framework for unit and integration testing.
- **Docker**: A containerization platform that allows the application to be packaged and deployed consistently across different environments.

- **Tailwind CSS**: A utility-first CSS framework for styling the application's user interface.

- **GitHub**: A version control system for managing source code and collaborating with other developers.

- **npm**: A package manager for installing and managing dependencies.

- **SSH Key (ed25519)**: A secure key pair used for authentication and secure communication between servers.

## 3. Development Practices

To maintain code quality and ensure smooth development, the following practices are recommended:

### 3.1 Git Commit Convention

Adhere to the conventional commit message format for all commits. This format follows the pattern: `***<type>: <description>***`. The `***<type>***` can be one of the following:

- `***feat***`: A new feature
- `***fix***`: A bug fix
- `***docs***`: Documentation changes
- `***chore***`: Regular maintenance and housekeeping
- `***test***`: Adding or modifying tests
- `***refactor***`: Code refactoring

### 3.2 Git Commit Hooks

Bypass the commit hooks during development to save time. However, make sure to follow the commit message convention mentioned above.

### 3.3 Testing

Write unit tests and integration tests using the ***Jest*** testing framework to ensure the reliability and stability of the application. Aim for a high code coverage percentage to minimize bugs and regressions.

## 4. Project Organization

### 4.1 Codebase Organization

Organize the codebase into logical directories and files. A suggested structure is as follows:

- **/src**
  - **/components** # React Native components
  - **/screens** # Application screens and navigation
  - **/services** # Integration with external services (Firebase, APIs,...)
  - **/state** # Redux store, actions, and reducers
  - **/utils** # Utility functions and helpers
  - **/tests** # Test files
- **/server**
  - **/controllers** # Express.js route controllers
  - **/models** # Data models and schemas
  - **/services** # Business logic and external service integrations
- **/docs** # Documentation files
- **/scripts** # Build, deployment, and other scripts
- **/config** # Configuration files

### 4.2 Task and Project Management

Use a task management tool like Notion to organize and track tasks. Assign tasks, set deadlines, and update progress regularly. Hold weekly meetings to discuss project status, challenges, and plan future tasks.

### 4.3 Daily Agenda

Update the daily agenda at the end of each day to record completed tasks, pending tasks, and any important notes. This helps keep track of progress and serves as a reference for future work.

## 5. Conclusion

This technical document provides an overview of the SmartEats application, including its technology stack, development practices, and project organization. Following these guidelines will help ensure smooth development, maintainable code, and successful project delivery.