

Recipt.me Project Architecture Document

Team 2 – April 2020

Table of Contents

- 1.0 Introduction
- 2.0 High Level Hierarchy
 - 2.1 Hierarchy Diagram
 - 2.2 Hierarchy Description
- 3.0 Components Classification
 - 3.1 Presentation Layer
 - 3.2 Controller Layer
 - 3.5 Record Layer
 - 3.6 Data Access Layer
 - 3.7 Database Layer

1.0 Introduction

Recipt.me is a website that allows users and food fans to communicate with each other by sharing recipes and rate them. Recipt.me distinguishes itself from the other food web-services for the openness and the wide choice that it offers product-wise. The project is going to expand itself in order to welcome food providers and producers.

2.0 High Level Hierarchy

2.1 Hierarchy Diagram

2.2 Hierarchy Description

The architecture system for the Recipt.me application is an n-tier architecture. This architecture system is designed to allow for proper information hiding, modular components, and single system dependencies. The abstraction of the presentation layer, and consequently the User Interface (UI). There are multiple layers between the Presentation Layer and the lowest level, due to the significant challenges present in the optimization and control of the Processes design. The Database layer is the lowest level in the hierarchy.

3.0 Components Classification

3.1 Presentation Layer

Purpose: To display forms, controls, images and sounds to the user to create a fluid and efficient user experience.

Specific Nature: The presentation layer will be in charge of displaying appropriate images, menus, and functions to the user. This layer provides an interface to enter data into specific forms. When a user clicks a menu on the GUI, the code corresponding to that event will be called.

Subcomponents: Image Viewer, Insert Data

- **Image Viewer** – The Image Viewer subcomponent is used when a user need to view one or more images of the recipes.
- **Insert Data** – The Insert Data subcomponent is responsible for the action of adding data about recipes, then this subcomponent push data to lower layer to enable database to store information about the recipes.

3.2 Controller Layer

Purpose: Processes and responds to events, typically user actions, and may invoke changes on the model.

Specific Nature: The controller layer in our program will be in charge of collect information received form the lower layer or from the higher layer. It is responsible near the interpretation of the gesture of the user at the presentation level and it is responsible about the data and record that comes from the bottom.

3.3 Record Layer

Purpose: This layer is in charge of containing the classes that strictly consist of data.

Specific Nature: This layer will be used to store User data, Landmark data, and Error data and Location data. These classes will only contain properties (variables) that describe each data type.

Associated Constructs: User Record, Recipe Record

- **User Record** – User Record will consist of only properties describing a user. This class will be static, meaning there is only one copy of this class in memory.
 - *ID* – Numeric type descriptor to identify a user record.
 - *Name* - Contains the name of the user, used to access the Recip.me platform. This will be of type string.
- **Recipe Record** – Recipe Record will consist of only properties describing a Recipe

- *ID* – Will hold the id of the recipe, to identify it. This property is of type numeric.
- *Name* – Will hold the name of the recipe. This property is of type string.
- *Category* – Will hold the category of the recipe. This property is of type string.
- *Ingredients* – Will hold the ingredients of the recipe. This property is of type numeric, because it is an external key.

3.4 Data Access Layer

Purpose: This layer is in charge of communicating to the database. This layer should handle all of the database transactions and connectivity.

Specific Nature: This layer will be in charge of communication to our database which will in turn lead to populating the record layer.

3.5 Database Layer

Purpose: This layer is in charge of storing data in persistent storage.

Specific Nature: This will be our database management system. There will store Recipes, Errors, and eventually Language Support.