System Design Document

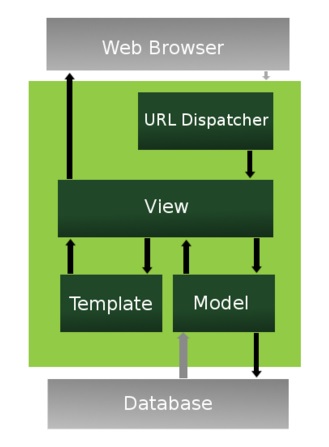
Project: Asian Food Center

By: Top-One team

# Architecture

## Web Framework

The solution will be built using a *Content Management System (CMS)* which provides the engine for the website, and allows a simple and fast way to develop a first working solution.

The diagram below gives an overview of one CMS architecture, for the Django framework, which is based on the Model-View –Controller pattern.

The CMS provides developers with templates for:

* The **models** (which are the entities to be stored in the database)
* The **views** (which represent the user interface or web pages that get displayed on the browser)

The CMS provides

* The **controller** through its framework (light green box) and the URL dispatcher

Developers will define the **Views** (based on templates) and the **Models** and the CMS will provide the rest.

Image Source: fishwarter (2009). The Django Web Application Framework. Retrieved from https://www.slideshare.net/fishwarter/the-django-web-application-framework-2-1221388

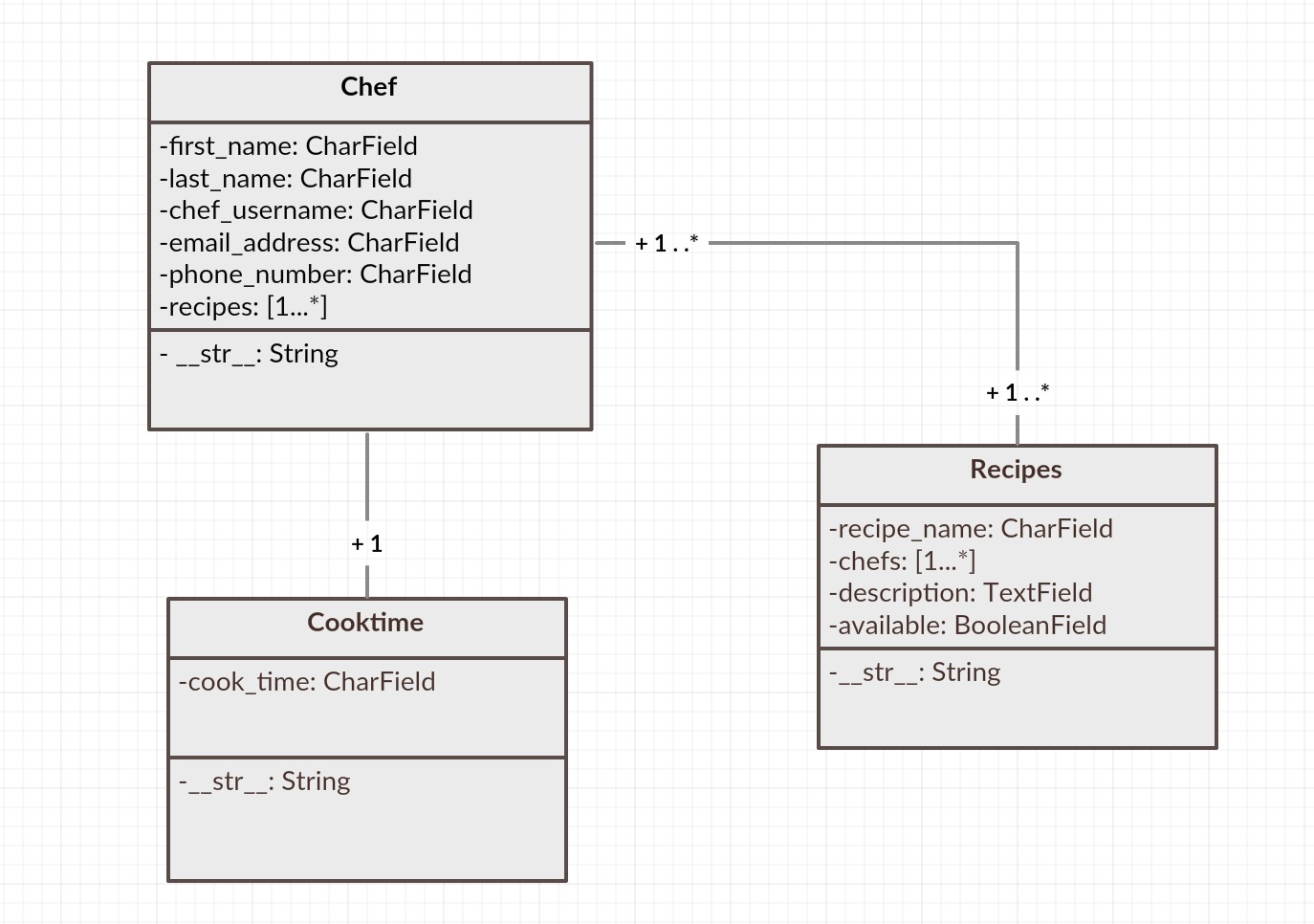
The sections below will document the Model and Views that are required for this application.

# Data Access Design

## Model Design (Data Model)

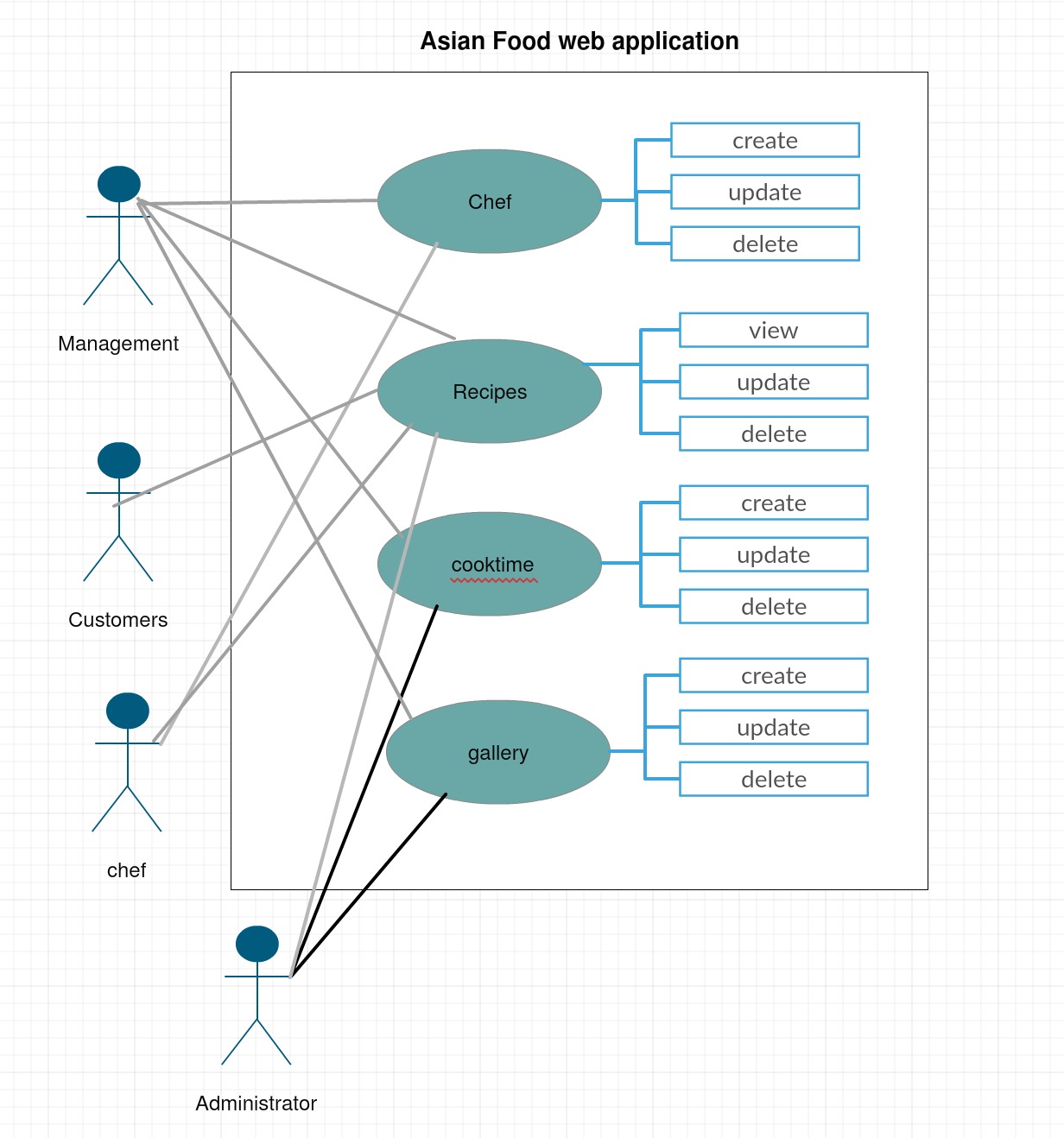
*This is the “model” part of the MVC pattern.*

*We refine the high level business domain model and create a detailed data model with all the entities that we will need to store in the database, with their attributes and relationships.*



## Functional Decomposition

*Select the key entities in the data model. From the user stories identify the types of functions needed on each entity, and which users will be performing them. This will provide you with a basis for identifying the Views that you will need.*

**

## Database Investigation

The Django framework will be used for this project. Django in its 'out-of-the-box' state is set up to communicate with SQLite which is a lightweight relational database included with the Python distribution. So by default, Django automatically creates a SQLite database for the project.

In addition to SQLite, Django *officially* supports (i.e. included in Django itself) three other popular relational databases that include: PostgreSQL, MySQL and Oracle. And *unofficially* (i.e. with third party packages) Django supports connectivity to other relational databases that include: SAP (Sybase) SQL Anywhere, IBM DB2 and Firebird, as well as the ADO (ActiveX Data Objects) and ODBC (Open Database Connectivity) interfaces, the last two of which are standard for connecting to Microsoft SQL Server and the latter is supported by most relational database brands.

# Security Design

## Framework Security

Which security features are provided by the web framework, which ones will you use

### User authentication

#### Authenticate all registered customers

It will be managed by the django framework system and record all the users information and secure it on the database.

#### authenticate all staff users

It will be provided by the django security management system. It will be configured so as to provide staff with user log-on, with the following user groups:

* Administrator (full access)
* Management team (create, update on Chefs)
* Chefs (create, update on recipes)
* General user(read only)

User IDs and passwords will be encrypted.

## Security Mechanisms

How will you implement each of the security mechanisms (if provided by framework, how will you use them, eg which settings, etc.)

Security requirements to be included:

* Prevent the system from most common security threats and attacks.
* Protect customer information.
* Use a secure payment process.
* Provide electronic order confirmation.

### Usability

* Application must use intuitive design, with easy navigation and functions for users to view the product and make a order.

### Portability

* The mobile phone interface is the first priority for customers.
* For staff users, the first priority is desktop and tablet.

## User Roles and Permissions

* Administrator (full access)
* Chefs (create, update on recipes)
* General user(read only)

User IDs and passwords will be encrypted.

# User Interface Design

### User Story 1

As a customer, I want to view products and services online quickly and easily.

* View: Recipes
* View: select recipes
* View: click recipe name
* View: view recipe description

### User Story 2

As an administrator, I want to be able to manage information so as to meet customer needs.

* Views: create recipes
* Views: update recipes
* Views: delete recipes
* Customer view/update

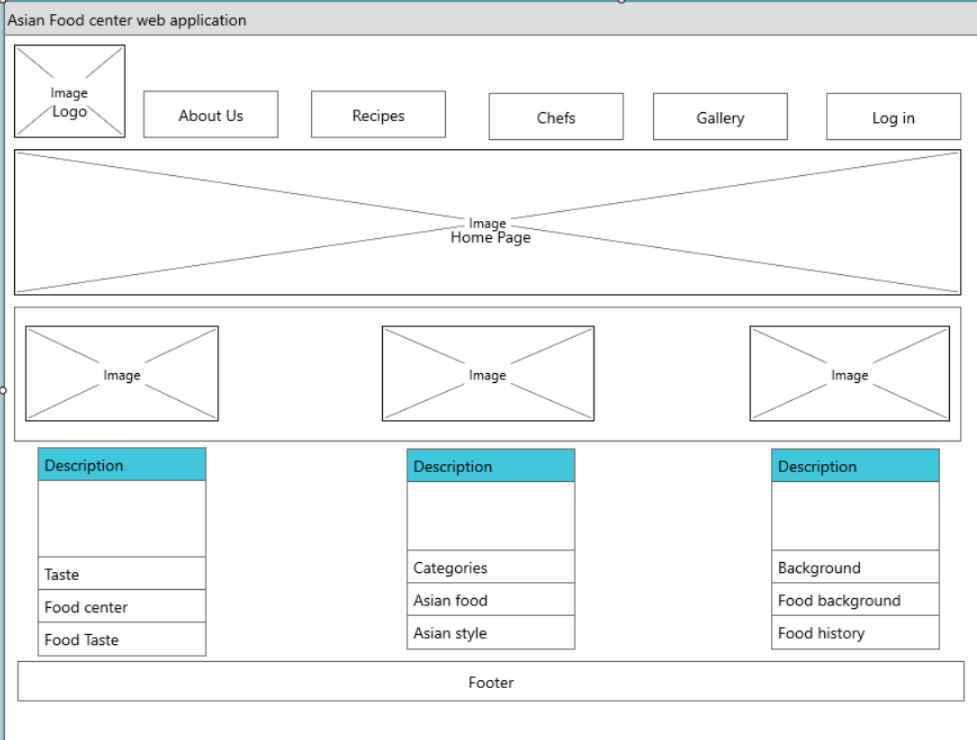
## UI Design

Record palette, fonts, look and feel, bootstrap used, etc. for your web app

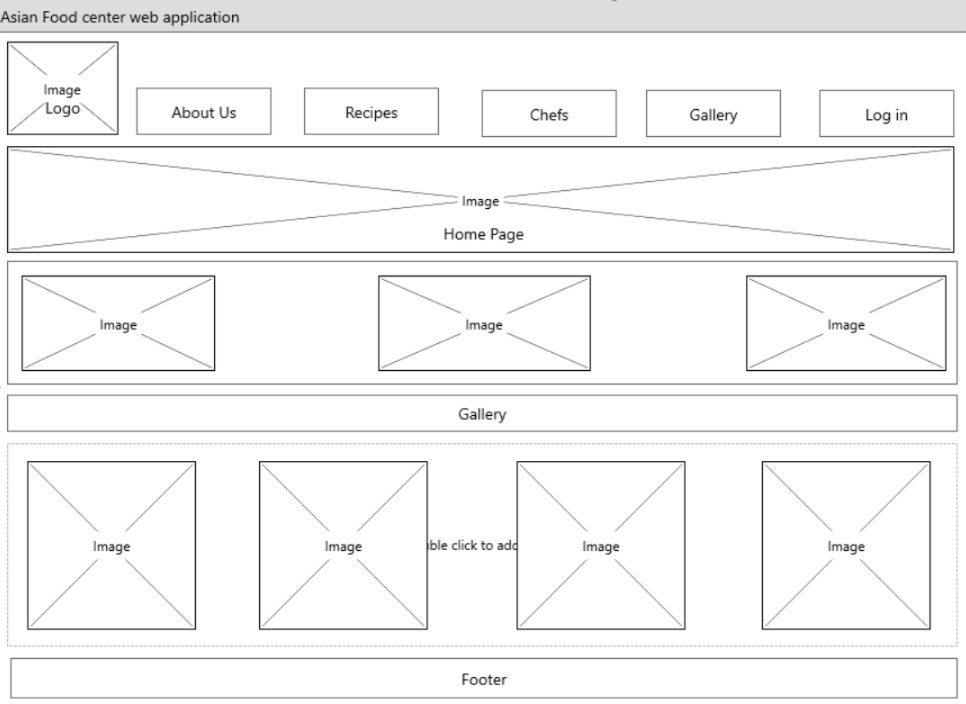
Include a wire-frame or mock-up for the landing page to illustrate the style you are aiming for.

Wire-frames for Asian Food Center web application:

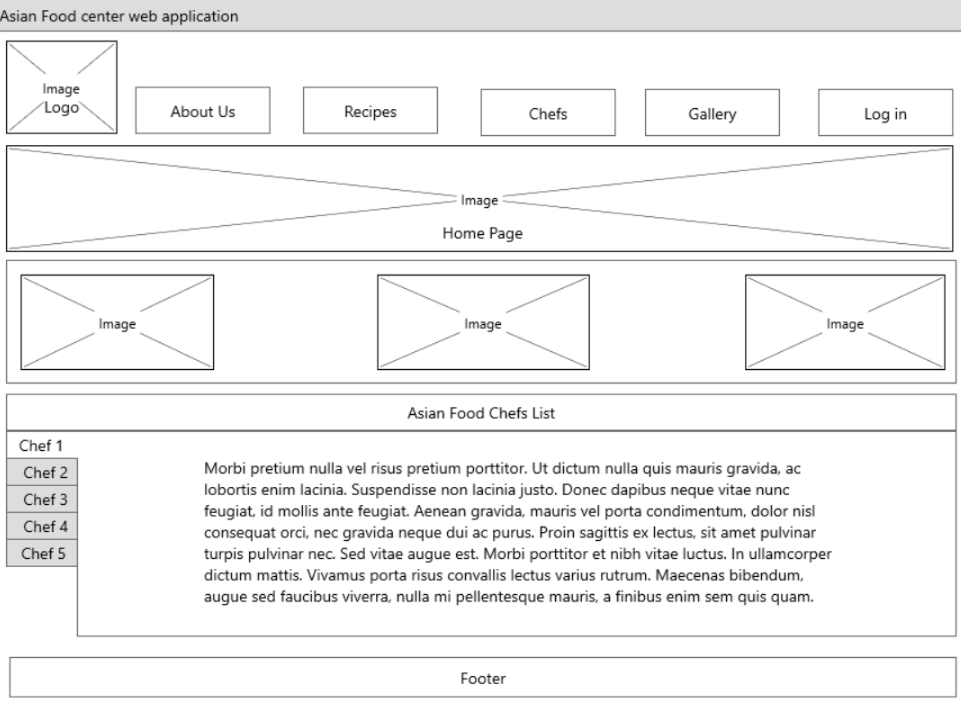
Homepage -



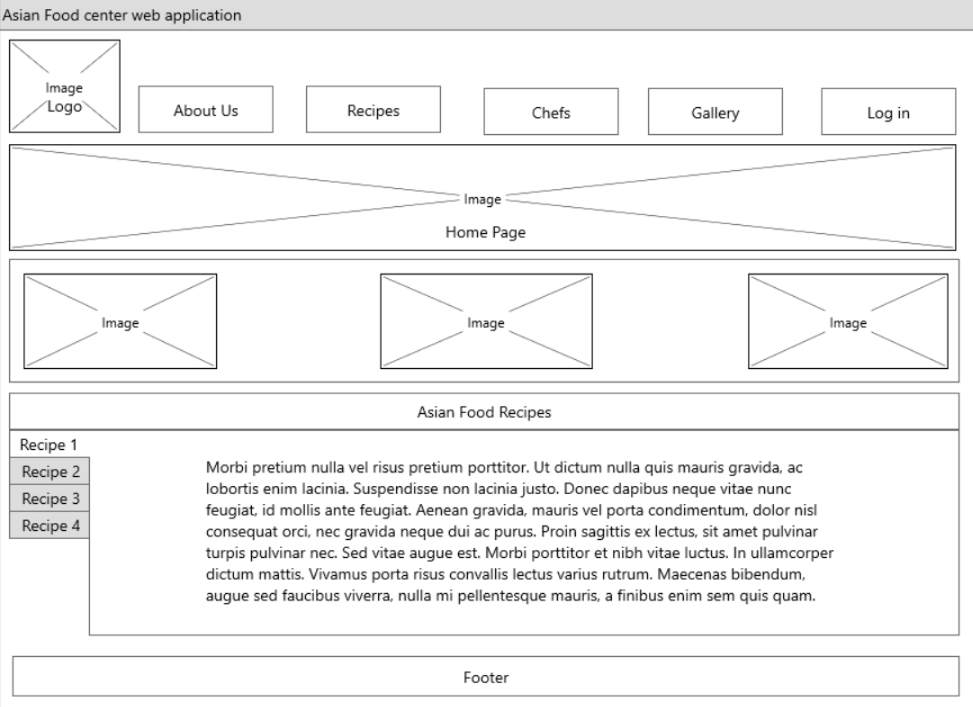
Gallery page -



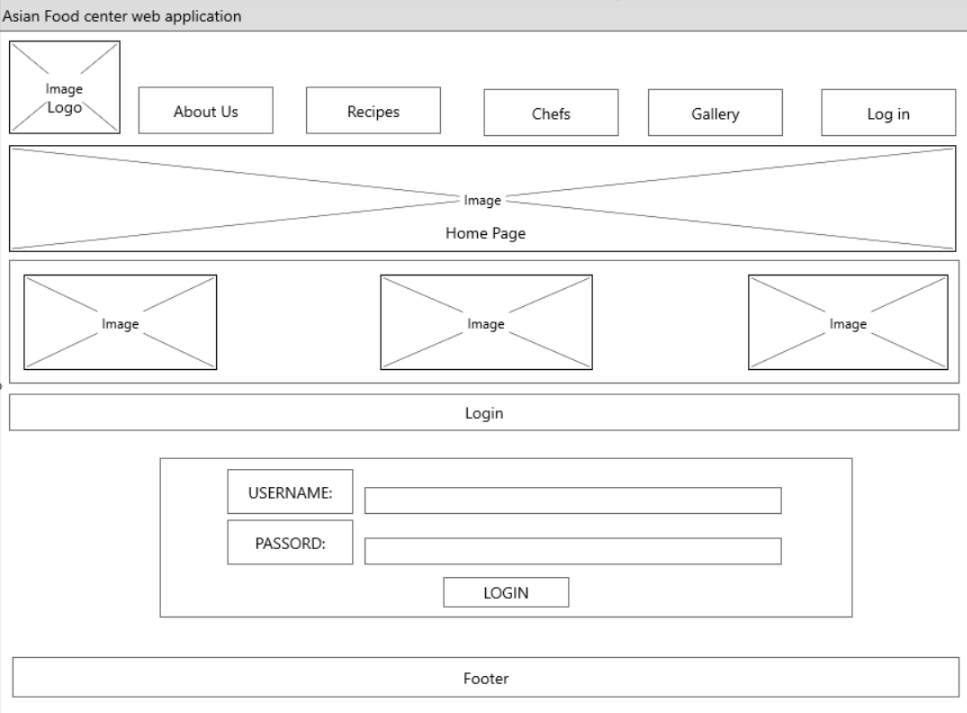
Chef page -



Recipe page -



Login page -



*This is the “view” part of the MVC pattern.*

*Overall site map will need:*

* *Homepage*
* *About us page*
* *Recipes page*
* *Chef page*
* *Gallery page*
* *Log on page*