**SYSTEMS ANALYSIS AND DESIGN**

**4.0 INTRODUCTION**

System analysis is a process of collecting factual data, understanding the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. System design is a phase where the new system is designed based on the user requirements and the detailed analysis of the existing system. It is the most crucial phase in the development of a system since the logical system design arrived at as a result of systems analysis is converted into physical system design.

This chapter will outline the general system and application architecture, requirements specification, functional and non-functional requirements, use case modelling, the database design, Entity Relationship diagram, Normalization, and User interface design.

**General System Architecture**

**General Application Architecture**

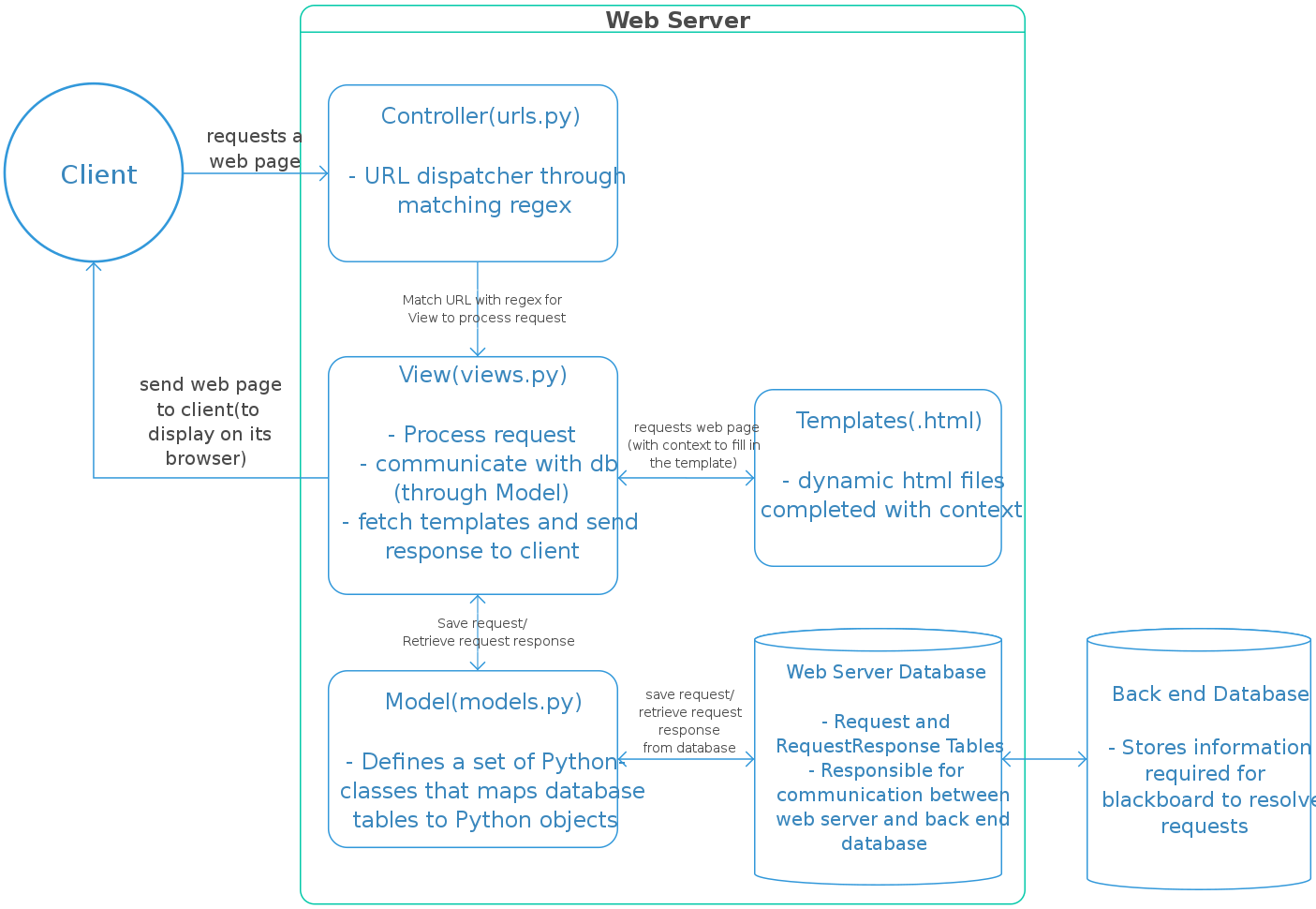
**The Model-View-Controller Design Pattern**

MVC has been around as a concept for a long time, but has seen exponential growth since the advent of the Internet because it is the best way to design client-server applications.

* The **model(M)** is a model or representation of your data. It’s not the actual data, but an interface to the data. The model allows you to pull data from your database without knowing the intricacies of the underlying database. The model usually also provides an *abstraction* layer with your database, so that you can use the same model with multiple databases.
* The **view(V)** is what you see. It’s the presentation layer for your model. On your computer,  
  the view is what you see in the browser for a Web app, or the UI for a desktop app. The view also provides an interface to collect user input.
* The **controller(C)** controls the flow of information between the model and the view. It uses programmed logic to decide what information is pulled from the database via the model and what information is passed to the view. It also gets information from the user via the view and implements business logic: either by changing the view, or modifying data through the model, or both.

Django follows the MVC pattern closely, however it does use its own logic in the implementation. Because the “C” is handled by the framework itself and most of the excitement in Django happens in models, templates and views, Django is often referred to as an *MTV framework*. In the MTV development pattern:

* **M stands for “Model,”** the data access layer. This layer contains anything and everything about the data: how to access it, how to validate it, which behaviors it has, and the relationships between the data.
* **T stands for “Template,”** the presentation layer. This layer contains presentation-related decisions: how something should be displayed on a Web page or other type of document.
* **V stands for “View,”** the business logic layer. This layer contains the logic that accesses the model and defers to the appropriate template(s). You can think of it as the bridge between models and templates.



**4.3 REQUIREMENT SPECIFICATION**

A system requirements specification is a comprehensive description of the intended purpose and environment for the system under development. What the software will do and how it will be expected to function is fully described under requirements specification. It outlines functional and non- functional requirements and may include a set of use cases that describe user interactions that the system must provide. Requirements specification allows a thorough analysis of requirements before design can start hence, reducing later redesign.

**4.3.1 FUNCTIONAL REQUIREMENTS**

Functional requirements describe in detail a system’s intended capabilities, appearance and interactions with users. It serves as a kind of guideline and continuing reference point as the system is being developed. The functional requirements for a system usually involves the user interface and describe each of the possible user input actions and the system’s response actions. Functional requirements therefore specify particular results of a system and it drives the application architecture of a system.

The system will have the following functional requirements:

* An administrator will be able to login and customize the system to the needs of the restaurant.
* Through the administrator’s panel, the system allows the manager to generate reports of orders, delete products and orders, and approve comments among others.
* The administrator can create multiple accounts for different users to perform specific actions like accepting orders, managing product stock etc.
* The administrator can check product inventory, accept orders and approve product reviews.
* Online Shoppers will be able to create an account so as to login to their portal and see their Order history etc.
* Online Shoppers will be able to place orders and receive notifications of their others in their emails.
* Online Shoppers can choose the language of their choice.
* Online Shoppers can review products with comments.
* Online Shoppers can also pay online before their orders are placed successfully

**4.3.2 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements essentially specify how a system should behave or work and also are a constraint upon the system’s behavior. They specify criteria that judge the operation of a system rather than specific behaviors. They also describe various attributes which affect the functionality’s effectiveness. Non-functional requirements do not alter a system’s functionality that is, the functional requirements remain the same regardless of the attributes attached to them. Non-functional requirements make up a significant part of the specification in that, users or clients may judge a system based on its non-functional requirements.

Below are the non-functional requirements our system will achieve:

* **Performance**

The system will be an interactive one hence the delays involved will be less in that there are no immediate delays in every action-response of the system.

* **Reliability**

As the system is meant for online shopping, the system will be reliable in that it will consistently perform according to its specification so as to enable users make place orders as and when they want to without any system failure.

* **Maintainability**

A system should be developed in such a way that it can evolve to meet the changing needs of a customer. The system will be developed in such a way that it can be customized to meet a particular vendor’s needs or preference.

* **Ease of use**

The system will ensure ease of use in that little training time will required to know how to use the system. The user interface will also be friendly so users will have ease when using the system.

* **Scalability**

It is the ability of a system to continue to function well when it is changed in size or volume to meet a user need. The system will be developed to fit onto any device such as a smartphone, laptop, tablet or desktop.

* **Security**

It is a system attribute that reflects the ability of a system to protect itself against external attacks, which may be deliberate or accidental. The system will be developed in such a way that proper login mechanism would be put in place to protect user accounts.

* **Safety**

The system will ensure safety so that information about orders placed is securely transmitted to the server without any changes in the information. PayPal Payment System is recognized internationally for it secure environment for online secured payments.

* **Hardware Requirements**

The hardware requirements are classified into server and client requirements which are detailed below:

* **Server Requirements**

Minimum CPU – Intel® processor Pentium IV minimum

Minimum Disk Space – 40GB

Minimum Memory – 2GB

* **Client Requirements**

Minimum CPU – Intel® processor Pentium IV 1GHz

Minimum Disk Space – 512MB

Minimum Memory – 256MB

* **Software Requirements**

The software requirement for the application runs best on a system with the following capabilities:

* Any operating system such as Windows, Linux, Macintosh, and Mozilla Firefox operating system
* A web browser such as Mozilla Firefox, internet explorer, opera, Google chrome etc.

**4.4 USE CASE MODELLING**

UML stands for Unified Modelling Language. They are diagrams to help aid the creation, visualization and documentation of various aspects of the software engineering process.

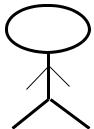
A use-case model is a model of how different types of users interact with the system to solve a problem. It therefore describes the goals of the users, the interactions between the users and the system and the required behavior in satisfying these goals.

**4.4.1 USE CASE DIAGRAMS**

A use-case diagram is used to graphically depict a subset of the model to simplify communications. It shows a subset of the model elements relevant for a particular purpose. Below is a table of use case symbols and their representations.

|  |  |
| --- | --- |
| SYMBOL | TERM AND REPRESENTATION |

**Actor:** An actor represents roles for users of a



system, including human users and other

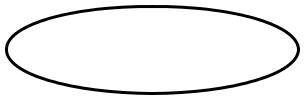
systems. An actor is external to a system.

**Association:** An association correspond to a

sequence of actions between the actors and use

case in achieving the use case.

**Use case:** A use case represents a user goal that



can be achieved by accessing the system.

**System Boundary:** The system boundary

defines the confines or scope of a system

within which the use cases are placed

Table 4.0 Use case symbols and their meanings.

The actors involved with the system are:

**Customer:** This actor is one of the main beneficiary of the system. The customer can order productsusing the system.

**Manager:** This actor is also another beneficiary of the system. Customizing the system to hispreference, editing products, approving orders and product reviews are some of the activities of this actor.

**System:** This actor is responsible for generating reports and sending confirmation messagesamong others. It however requires the administrator inputs to generate the reports.

**Use Case Scenarios**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Register | |
|  |  | |
| Actor | Customer | |
|  |  | |
| Description | Enables new customers to create an account by | |
|  | entering their credentials. | |
|  |  |  |
| Steps Performed | 1. | Click on sign up |
|  | 2. | Enter personal details and password. |
|  | 3. | Click on sign up button. |
|  |  |  |

Table 4.1 Customer Registration

|  |  |  |
| --- | --- | --- |
| Use Case Name | Login |  |
|  |  | |
| Actor | Customer | |
|  |  | |
| Precondition | Customer should have already registered. | |
|  |  | |
| Description | Enables existing customers to have access to | |
|  | the system’s functionality using valid | |
|  | credentials. | |
|  |  |  |
| Steps Performed | 1. | Click on login |
|  | 2. | Enter email and password. |
|  | 3. | Click on login. |
|  |  |  |

Table 4.2 Customer login

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | View Menu | | |
|  |  | |  |
| Actor | Customer | | |
|  |  | |  |
| Description | Enables a customer to view products available | | |
|  | together with their prices. | | |
|  |  | |  |
| Steps Performed | Go to the website of the using a web | | |
|  | browser. You can also view products after login to the dashboard. | | |
|  |  | | |
|  |  |  |  |
| Table 4.3 View Products |  |  |  |
|  |  | |  |
| Use Case Name | Make Order | |  |
|  |  | |  |
| Actor | Customer | |  |
|  |  | |  |
| Precondition | Customer can login or choose not to login. | |  |
|  |  | |  |
| Description | Enables customers to order products which is | |  |
|  | successful after they have proceeded with the | |  |
|  | payment. | |  |
|  |  |  |  |
| Steps Performed | 1. | Enter quantity |  |
|  | 2. | Click on add to cart button to add |  |
|  |  | product to cart. |  |
|  | 3. | Click on checkout. |  |
|  | 4. | Enter confirmation and payment details |  |
|  |  | and click on Place now. |  |
|  | 5. | Proceed with payment using either |  |
|  |  | debit card. |  |
|  |  |  |  |

Table 4.4 Placing Orders

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Use Case Name | Login |  |
|  |  | |
| Actor | Manager | |
|  |  | |
| Description | Enables a manager to access the system’s | |
|  | functionality available to perform managerial | |
|  | duties. |  |
|  |  |  |
| Steps Performed | 1. | Proceed to admin Panel. |
|  | 2. | Enter email and password. |
|  | 3. | Click on login. |
|  |  |  |

Table 4.5 Manager Login

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | |
| Use Case Name | Edit Menu | |
|  |  | |
| Actor | Manager | |
|  |  | |
| Description | Enables a manager to edit available menus | |
|  | such as adding new menu, deleting menu not | |
|  | available as well as upload images for the | |
|  | various menu. | |
|  |  | |
| Precondition | Manager should login. | |
|  |  |  |
| Steps Performed | 1. | Click on menus. |
|  | 2. | Make changes to the menu by adding |
|  |  | or deleting. |
|  |  |  |

Table 4.6 Editing Products

|  |  |  |
| --- | --- | --- |
| Use Case Name | Edit menu category | |
|  |  | |
| Actor | Manager | |
|  |  | |
| Description | Enables the manager to group menu based on | |
|  | its category such as continental or local dishes. | |
|  |  | |
| Precondition | Manager should login. | |
|  |  |  |
| Steps Performed | 1. | Click on categories |
|  | 2. | View all categories. |
|  | 3. | Make the necessary changes by adding |
|  |  | or deleting. |
|  |  |  |

Table 4.7 Edit Product Category

|  |  |  |
| --- | --- | --- |
| Use Case Name | Generate Report | |
|  |  |  |
| Actor | System |  |
|  |  | |
| Description | Generates reports on periodic basis. | |
|  |  |  |
|  | 1. | Click on report on the manager’s |
| Steps Performed |  | interface. |
|  | 2. | Enter the period, that is start and end |
|  |  | date desired for report. |
|  |  |  |

Table 4.8 Report Generation

|  |  |
| --- | --- |
| Use Case Name | Send SMS/Email |
|  |  |
| Actor | System |
|  |  |
| Description | Sends order confirmation to the customer and |
|  | order details to the vendor via Email. |
|  |  |
|  | When an order is successful after payment, the |
| Steps Performed | system automatically sends an email |
|  | to the customer and vendor. |
|  |  |

Table 4.9 Send SMS/Email

**4.4.2 SEQUENCE DIAGRAM**

Sequence diagrams are the most common kind of interaction diagrams that shows how actors and objects interact to realize a use case scenario. It focuses on the message interchange between a number of lifelines. We normally draw a sequence diagram if we have a use case, to describe how the main components of the system interact. Then again sequence diagram helps us identify messages arriving at an interface of a component, to describe how the internal parts of the component interact.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| SYMBOL | | | | | TERM & MEANING |
|  |  |  |  |  |  |
|  |  |  |  |  | **An actor**: Is an entity or system that derives |
|  |  |  |  |  | benefit from and is external to the system. |
|  |  |  |  |  |
|  |  |  |  |  | Participates in a sequence by sending and/or |
|  |  |  |  |  | receiving messages. |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | **An object**: Participates in a sequence by |
|  | An Object: a class | | |  |
|  |  | sending and/or receiving messages. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | **A lifeline**: Represents the life of an object during a sequence. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | **A focus of control**: Is a long narrow rectangle |
|  |  |  |  |  |
|  |  |  |  |  | placed on or above a lifeline. Denotes when an |
|  |  |  |  |  | object is sending or receiving messages. |
|  |  |  |  |  |  |
|  | **A Message ()** | | | | **A message**: Conveys information from one |
|  |  |  |  |  | object to another. |
|  |  |  |  |  |  |
|  | X | | | | **Object destruction**: An X is placed at the end |
|  |  |  |  |  | of an object’s lifeline to show that it is going |
|  |  |  |  |  | out of existence |
|  |  |  |  |  |  |



Table 4.13 Sequence diagram symbols and their meanings