

# Fast n' Delight Cafe

## Phase 3

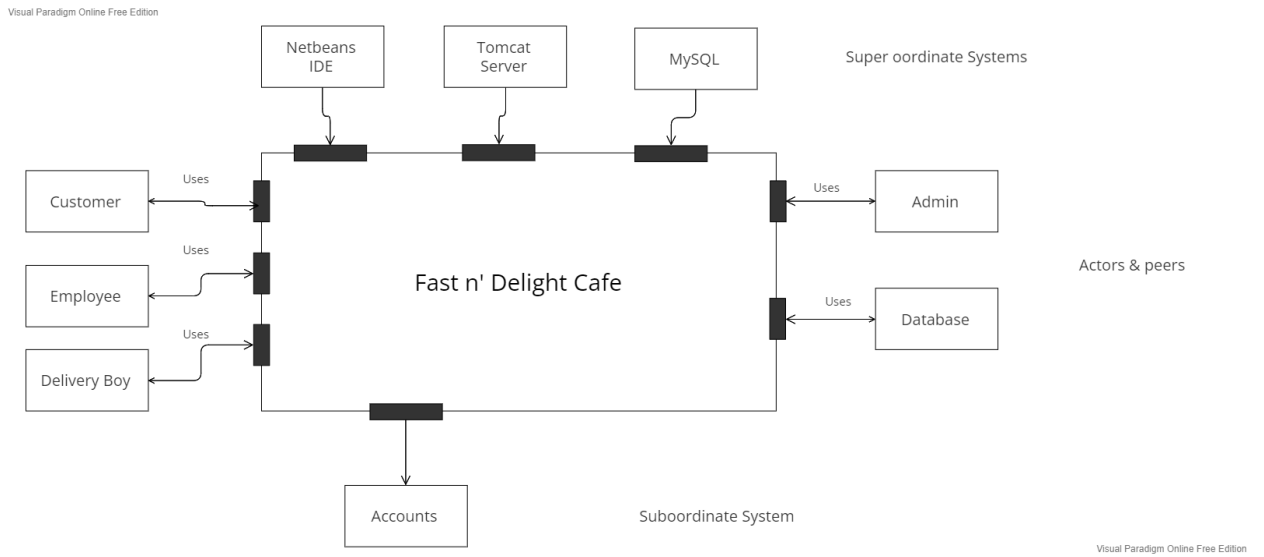
## Team 2

| Member Name    | Member Roll # | Primary Responsibility  |
|----------------|---------------|---|
| Usman Faisal   | 20L-1385      | SAD, UC-1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 |
| Fizza Sajjad   | 20L-1389      | SAD, UC-17, 18, 19  |
| Noor Fatima    | 20L-0993      | SAD, UC-6, 8, 12  |
| Abdur Rafay    | 20L-1391      | SAD   |
| Fiza Hassan    | 20L-1279      | SAD   |
| Hassan Murtaza | 20K-1641      | SAD, UC-6, 8, 12  |
| Hira Touqeer   | 20L-1375      | SAD   |

## **Table of Contents**

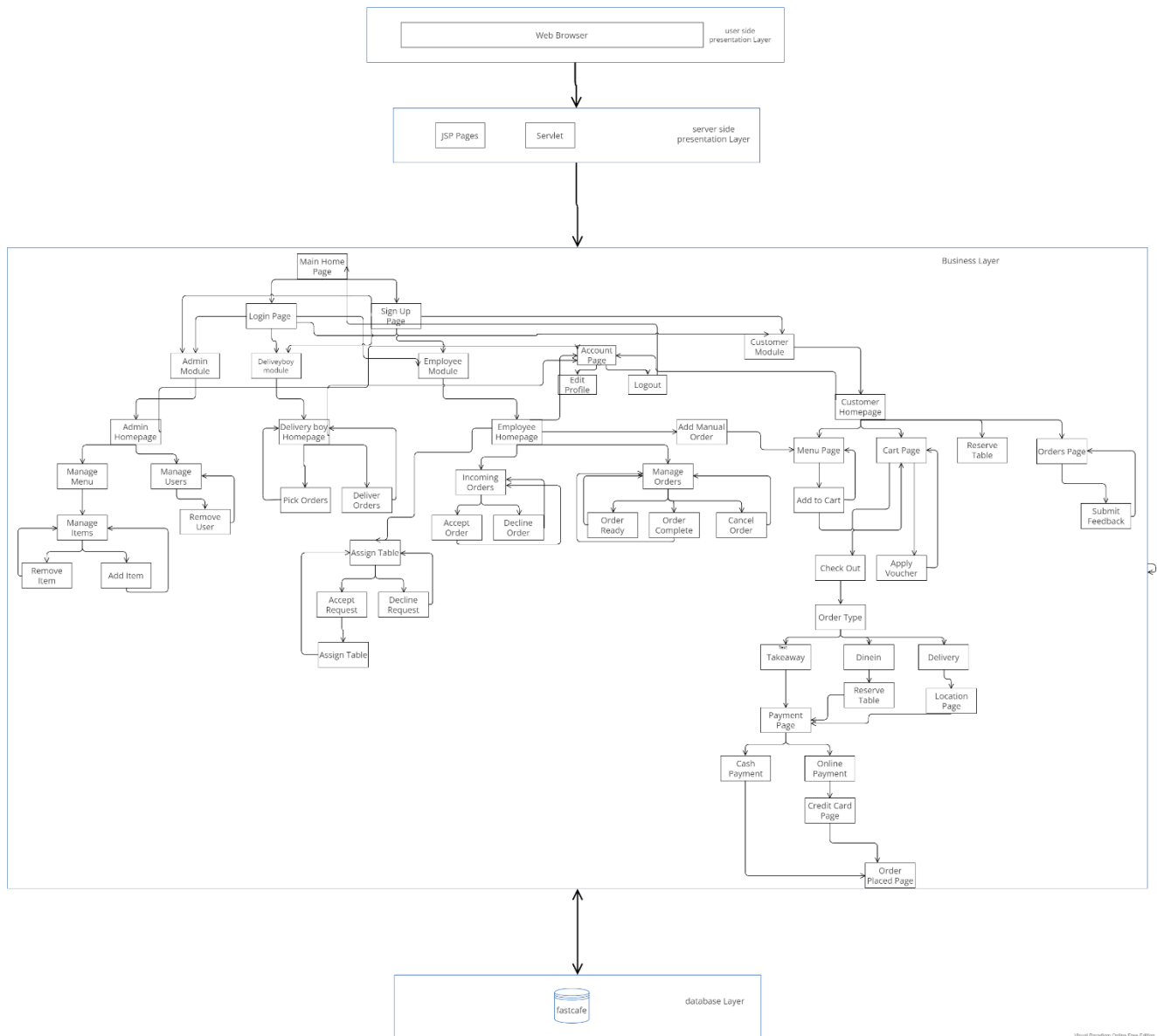
|                                      |          |
|--------------------------------------|----------|
| <b>Architectural Context Diagram</b> | <b>2</b> |
| <b>Architectural Style</b>           | <b>3</b> |
| <b>Rationale</b>                     | <b>4</b> |

# 1. Architectural Context Diagram



## 2. Architectural Style

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

### 3. Rationale

| Serial # | Component Name         | Component description  | Reference Use Case |
|----------|------------------------|--|--------------------|
| 1        | Home Page              | The main home page of the site, users can login/signup from here |                    |
| 2        | Signup Page            | The actors can sign up to their respective portal.               | UC-1               |
| 3        | Login Page             | The actors can sign up to their respective portal.               | UC-2               |
| 4        | Log out                | Actors can logout from system.                                   | UC-3               |
| 5        | Edit Profile           | Actors can edit their profile                                    | UC-4               |
| 6        | Admin Home Page        | Admin's main menu  |                    |
| 7        | Manage Menu            | Admin can manage menu of the cafe                                | UC-17              |
| 8        | Manage Items           | Admin can manage items of the cafe                               | UC-17              |
| 9        | Edit Item              | Admin can edit items of the cafe                                 | UC-17.3            |
| 10       | Add Item               | Admin can add new items  | UC-17.1            |
| 11       | Order Details          | Users can view order details.                                    | UC-15              |
| 11       | Manage Users           | Admin can manage users of the cafe                               | UC-18, UC-19       |
| 12       | Remove Users           | Admin can remove users of the cafe                               | UC-18.2, UC-19.2   |
| 13       | Delivery Boy Home Page | Delivery boy's main menu   |                    |
| 14       | Pick Order             | Delivery boy can pick orders from cafe. //Not Implemented        |                    |
| 15       | Deliver Order          | Delivery boy can deliver orders in university. //Not Implemented |                    |
| 16       | Employee Home Page     | Employee's main menu   |                    |
| 17       | Incoming Order         | Employees gets new order in his portal                           | UC-6               |

|    |                  |  |                   |
|----|------------------|--|-------------------|
| 18 | Accept Order     | Employee can accept the new order                    | UC-6              |
| 19 | Decline Order    | Employee can decline the new order                   | UC-6              |
| 20 | Manage Orders    | Employee can manage the status of orders             | UC-9,UC-16, UC-15 |
| 21 | Cancel Order     | Employee can cancel the order within 10 minutes      | UC-9              |
| 22 | Order Complete   | Employee can mark the order as complete or delivered | UC-16             |
| 23 | Order Ready      | Employee can mark the order as ready                 |                   |
| 24 | Add Manual Order | Employee can add a manual order                      | UC-12             |
| 25 | Assign Table     | Employee can assign table to customer request        | UC-8              |
| 26 | Menu Page        | Shows the list of available items                    | UC-11             |
| 27 | Add to cart      | Customer can items to cart                           | UC-5              |
| 28 | Cart Page        | Customer can check out or use voucher                | UC-5, UC-14       |
| 29 | Apply Voucher    | Customer can avail voucher                           | UC-14             |
| 30 | Check out        | Customer can move to complete their order            | UC-5              |
| 31 | Order Type       | Customer can choose type of order delivery method    | UC-5              |
| 32 | Takeaway         | Customer chooses takeaway method                     | UC-5              |
| 33 | Dinein           | Customer chooses dinein method                       | UC-5              |
| 34 | Delivery         | Customer chooses delivery method                     | UC-5              |
| 35 | Reserve table    | Customers can reserve a dine in table                | UC-7              |
| 36 | Location Page    | Customer chooses location of their order delivery    | UC-5              |

|    |                   |  |             |
|----|-------------------|--|-------------|
| 37 | Payment Page      | Customer chooses payment method                | UC-5, UC-13 |
| 38 | Cash Payment      | Customer chooses cash payment method           | UC-13       |
| 39 | Online Payment    | Customer chooses payment method                | UC-13       |
| 40 | Credit Card Page  | Verifies the entered credit card of customer   | UC-13       |
| 41 | Order Placed Page | Shows the orders details                       | UC-5, UC-15 |
| 42 | Vouchers Page     | Customers Can view list of vouchers            | UC-14       |
| 43 | Orders Page       | Customers Can view their orders                | UC-15       |
| 44 | Submit feedback   | Customers Can submit feedback for their orders | UC-10       |
| 45 | View feedback     | Admin Can view feedback for orders             | UC-22       |

Architectural Style Used: Layered + Call & Return

WebApps 8 are client-server applications typically structured using multi layered architectures, including a user interface or view layer, a controller layer which directs the flow of information to and from the client browser based on a set of business rules, and a content or model layer that may also contain the business rules for the WebApp. The user interface for a WebApp is designed around the characteristics of the web browser running on the client machine (usually a personal computer or mobile device). Data layers reside on a server. The architectural design of a WebApp is also influenced by the structure (linear or nonlinear) of the content that needs to be accessed by the client. The architectural components (Web pages) of a WebApp are designed to allow control to be passed to other system components, allowing very flexible navigation structures. The physical location of media and other content resources also influences the architectural choices made by software engineers.

### **Advantages of using layered architecture**

#### **Ease of development:**

Developers don't need full knowledge of the project. Different people have different skills. They can develop applications by separating skill sets by layers. (e.g. if you are good at the front end, you can develop a presentation layer. If you have skills at backend or database, you can develop a business layer or database layer.

#### **Testability:**

A developer can mock a presentation component or screen to isolate testing within a business component, as well as mock the business layer to test certain screen Functionality.

**Maintainability:**

The layers of isolation concept mean that changes made in one layer of the architecture generally don't impact or affect components in other layers: the change is isolated to the components within that layer Abstraction. Layered architecture abstracts the view of the system as a whole while providing enough detail to understand the roles and responsibilities of individual layers and the relationship between them.

**Encapsulation:**

No assumptions need to be made about data types, methods and properties, or implementation during design, as these features are not exposed at layer boundaries. Clearly defined functional layers. The separation between functionality in each layer is clear. Upper layers such as the presentation layer send commands to lower layers, such as the business and data layers, and may react to events in these layers, allowing data to flow both up and down between the layers.

**High cohesion:**

Well-defined responsibility boundaries for each layer, and ensuring that each layer contains functionality directly related to the tasks of that layer, will help to maximize cohesion within the layer. (business layer has the all logical application.

**Low coupling:**

Communication between layers is based on abstraction and events to provide low coupling between layers.

**Reusable:**

Lower layers have no dependencies on higher layers, potentially allowing them to be reusable in other scenarios.

## Disadvantage

**Performance Degraded**

While it is true some layered architectures can perform well, the pattern does not lend itself to high-performance applications due to the inefficiencies of having to go through multiple layers of the architecture to fulfill a business request.

**Not Easy to deploy:**

Depending on how you implement this pattern, deployment can become an issue, particularly for larger applications. One small change to a component can require a redeployment of the entire application (or a large portion of the application) resulting in deployments that need to be planned, scheduled, and executed during off-hours or on weekends. (Because the whole running applications have to stop to rebuild from 0) As such, this pattern does not easily lend itself toward a continuous delivery pipeline, further reducing the overall rating for deployment.

**Low Scalability:**

Because of the trend toward tightly coupled and monolithic implementations of this patterns, applications built using this architecture pattern are generally difficult to scale. You can scale a



layered architecture by splitting the layers into separate physical deployments or replicating the entire application into multiple nodes, but overall the granularity is too broad, making it expensive to scale.