Contents

Architectural Requirements

Introduction

The system design to be implemented will be referred to as a Hybrid-tier System. This hybrid model is based off the n-tier (multi-layered) architecture, where the system is divided into layers. This multi-layered system is also referred to as a Client-Sever architecture. For this architecture, there are four tiers/layers. Those being:

1. Presentation Tier
2. Server Tier
3. Database Tier
4. AI (Artificial Intelligence) Algorithm Tier

Each layer will be modifiable without needing to change the entire application. As a result of the aforementioned, maintenance and adding extra functionality is easier. Overall complexity of code over all layers will be reduced, thus making the layers reusable in other applications. Since the layers do not act like a traditional n-tier where one would delegate tasks to other layers; once a request goes to the server tier, the server will decide if it will go to the database tier or the AI tier which then goes to the database tier. Thus showing an optimised parallel system to deal with requests. Additionally, the use of this architecture system will allow different members of our team to modify separate layers without interference. It is possible to deploy each layer (specifically the database and server layers) over multiple locations for better reliability and performance. What differentiates the hybrid architecture from a traditional n-tier or client-server, is in how the layers are implemented. Figure 1 shows the overall architecture of this hybrid system.

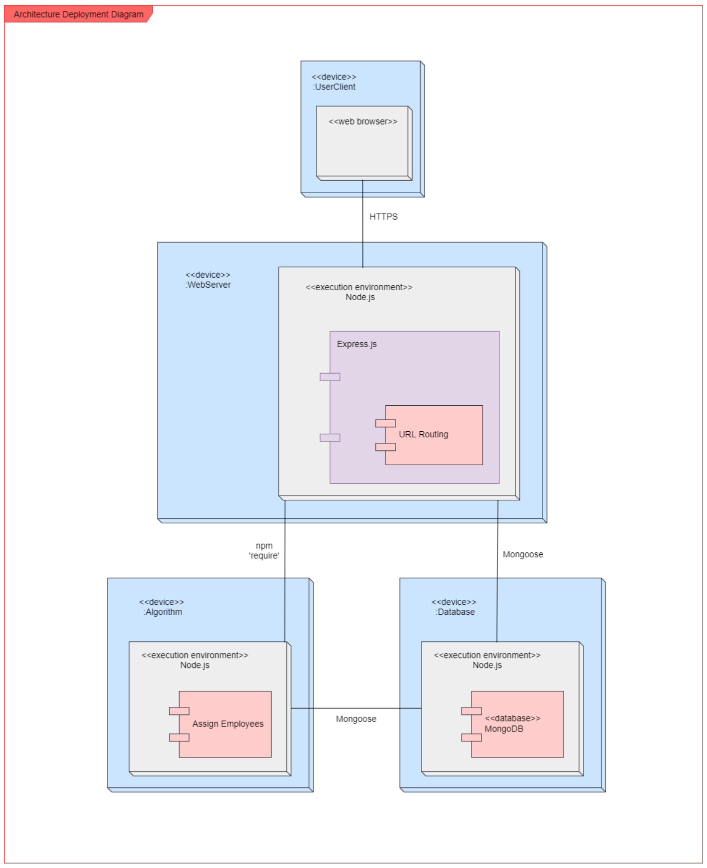

Figure 1: Deployment Diagram: Hybrid-Tier Architecture

Presentation Tier

This is the layer the user will be interacting with. It encompasses the actual applications that can be developed (Android, iOS and web), and any interactions that the users may have with them. In this system, the user will primarily work with a web application as the front-end.

The presentation tier includes:

- Displaying the user interface (UI), which the user will interact with.
- Adding notification, forms, buttons and pages.
- Providing an interface to trigger requests to the system.
- Ultimately serving as a communication medium between pages on the front-end, and to send and receive data between the back-end.

Server Tier

The server layer implements the business logic, and process all requests made by the front-end - it makes logical decisions based on the interactions from the presentation layer and then decides what or where data needs to go; either to the database layer or the AI layer.

The server tier is responsible for:

- Allow admin users to be able to add/remove/edit client projects, and assign employees to specific projects.
- Automatically update calendars of employees.
- Crosscheck projects before assigning them to an employee.
- Server communication over the internet.
- Scheduling Assistant calculations.
- Limit an employee from being able to edit projects they have been assigned to.
- Allow an employee to add other events to his/her calendar if these do not clash with the pre-populated events.
- Session management.

Database Tier

The data layer is where the information used by the presentation layer for sending and retrieving, is stored. This layer consists of:

- Recommended time slots where all employees are available.
- Server storage with MongoDB
- View individual employee calendar and assigned projects.

AI (Artificial Intelligence) Algorithm Tier

The AI layer houses the resource allocation algorithm, and is used by the server layer to allocate employees to a project. This layer runs off its own independent resources, and is dependent only on the database tier from which it retrieves resources necessary for the allocation algorithm.