High Level Design: Eden Fresh

Provided by Team Web Crawlers

Alexander Dung

Devarsh Patel

Justin Calma (Team Leader)

Kayla Chu

Sze Man Tang

Submission Date: October 6, 2021

1. Introduction

1.1. Purpose

HLD document provides the brief information of a comprehensive architecture overview related to EdenFresh. The HLD document represents technical aspects of the tool systems components and is intended to convey and capture the significant architectural design decisions which have been made in preparation of the web applications called EdenFresh. The HLD will be focused on the 3-tier architecture and microservices architecture design. 3-tier architecture will separate the different components in layers. Microservices architecture or microservices is a development approach in which single applications are composed of independently deployable smaller services.

1.2. Scope

The scope of the HLD presentation to structure the base level for creating a path to understand the systems of interactions between database, application flow, Business logic, simple logic, input validation and error handling for the front and back end framework of the EdenFresh web applications. The HLD will separate the connections between services and components using 3-tier and microservices which will be divided as rows and columns. Microservices scope is when one server will make any changes or updates it will not affect the other servers. 3-tier or layers scope separates the components which separates the work independently between 3 different web developers.

2. HLD Hardware and Software

The HLD system architecture design has two different main components that separates the User Interface between Hardware and Software. Users most of the time will interact with EdenFresh web applications that will be created as software. Hardware will be most lickly physicals tools that supports the logical interact between users and software. HLD hardware and software will have a front end and back end framework. (Figure 1)

2.1. Hardware Architecture

The hardware will come in categories which contain the tools such as client, browser, web server, external server and database storage. These are the tools in components systems that make visible web applications in the Hardware.

Front End: The front end systems selected by the users where users will be able to interact with functional and nonfunctional requirements. There are three components that fit into the front end such as client, browsers, and web applications.

- Client: is a physical tool used by the users such as laptop, mobile and tablet.
- Browsers: are pieces of software that access and display pages, files on the web. EdenFresh users will be able to access the website through a browser using any client. Browsers will be used in EdenFresh Google *Chrome for Windows 10* 89+.
- Web Applications: EdenFresh will be the page of web applications that users can interact with through the google chrome browsers.

Back End: The Back end systems take the request from front end components and execute with back end components. There are two tools used for hardware back end frameworks such as web server and database.

- Web server: it will take requests from the web applications and in a few seconds of processing it will respond back to the web applications. There will be

communications using the HTTP server between web server (back end) and web applications (front end) for sending a request and getting a response.

- Database: will store all the details of users, sales, products and chat conversations until the user wants to change or update it. There will be communications between the web server and database for sending a request and getting a response.

External System: will include another server by developer to keep web applications quick access to different sources.

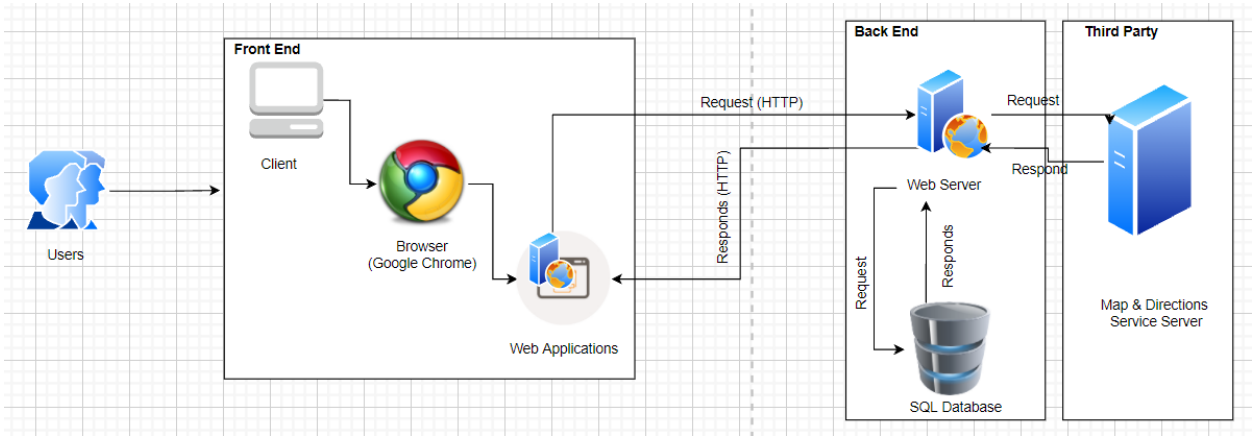

Figure.1 Hardware Architecture HLD

2.2. Software Architecture

The software will come in categories which use a simple application page (SPA) that communicate using MVVM design from front end to back end framework. MVVM is model-view-view-model. MVVM sends the request and gets a response between view to viewmodel and viewmodel to model. Software architecture will be separated into two different parts such as 3-tiers or layers architecture and Microservices architecture. (Figure 2)

Front End: The front end systems selected by the users where users will be able to interact with functional and nonfunctional requirements. Front end development is most likely

client side server that communicates between View(user can see) and View Model (get job done).

- Client: is a physical tool used by the users such as laptop, mobile and tablet.
- View: This is the only part of the application that the users actually interact with. It contains the data bindings, events and behaviors. It is an interactive UI that represents the state of a ViewModel. View isn't responsible for handling state- it keeps this in sync with ViewModel.
- ViewModel: can be considered as a specialized Controller that acts as a data converter. It changes Model information into view information, passing commands from View to Model. ViewModel stands between view and model. View model comes in the presentation layers of the applications.
    1. Simple Business Logic is basically all features interaction by the users provided in the web applications. The features could be chat, reviews, transactions etc.
        a. Input Validation will make sure that the user's input will be accurate as needed which is set by business logic (backend). Sometimes, it just passes the input to server side applications(back end) to check further validation exceptions.
        b. Error Handling: error handling will tackle actions or solve immediate error issues instead of showing the user that something went wrong.
    2. Security in client side server to make sure privacy of user interactions during users use.

Back End: Back end development refers to the server side of development where you are primarily focused on how the site works. Server side applications will take all the requests from the client side server and execute each request after each request passes the validation and error based components. Below tools come into the application layer that separates applications tier and different bases services using microservices.

API Gateway: The API gateway is the entry point for clients. Instead of calling services directly, clients call the API gateway, which forwards the call to the appropriate services on the back end.

- Application logic: The user's interactions with each application's features. Application logic handles everything else besides business rules.
    - Input validation: will check if all the inputs are valid or not and send the response back with results that the user requested is approved or not.
    - Error Handling: takes place to stop inconvenient errors happening in the applications logic.
- Security: will make sure the privacy data stay secure during user interaction protocols.
- Complex Business logic: will have business level collaborations, agreements and rules where users can access limited tools at limited supports in different categories features in EdenFresh. For example, users will have limited chatting characters.
    - Input validation: will check if all the inputs are valid or not by the business rules and send the response back with results that the user requested is approved or not.
    - Error Handling: take place to stop inconvenience errors happening in the business logic.

- Login: will check all the users login access if it is validated or has an error which will make sure the right users are accessing the right database.
    - Input validation: will check users login information input if it is valid or not.
    - Error Handling: take place to stop inconvenience errors happening during login.
- Update: will take place when client side servers request to update anything in their web applications features.
    - Input validation: will check if all the updated information is acceptable to update or not by validations.
    - Error Handling: take place to stop inconvenience errors during any update information.

Database Layer: Database Layers represent DBMS design which will update all existing data and insert all the new data that enter from client side server. We will be using SQL database server.

Third Party Layer: will include the external server for the EdenFresh to access map and directions service server.

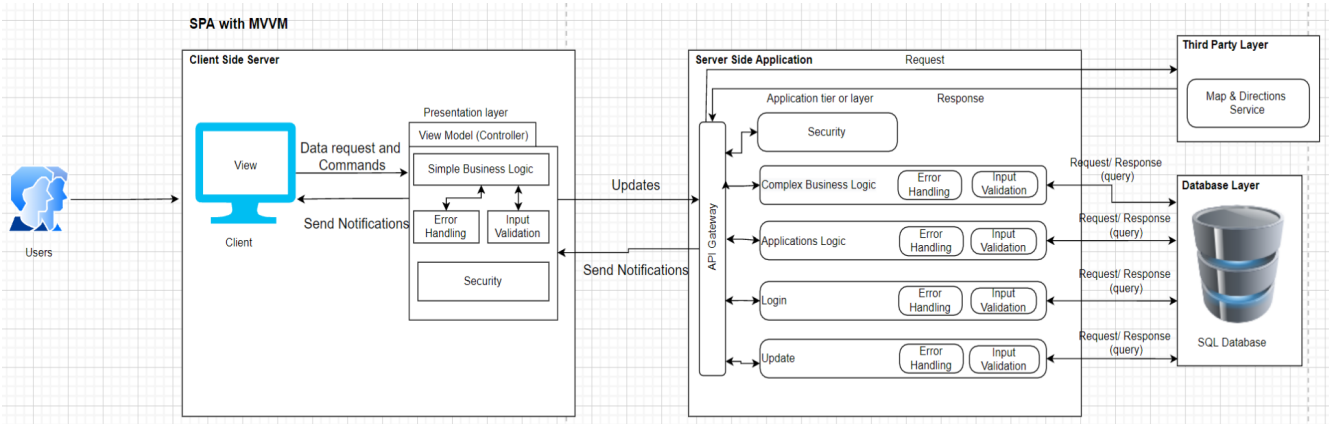

Figure.2 Software Architecture HLD