Courses Roadmap

Software Architecture Document

Version 1.2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/11/2021 | 1.0 | Create Software Architecture | Ngo Huy Anh |
| 23/11/2021 | 1.1 | Update Use-case model | Ngo Huy Anh |
| 25/11/2021 | 1.2 | Update Software Architecture | Ngo Huy Anh |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Architectural Goals and Constraints 4

3. Use-Case Model 4

4. Logical View 4

4.1 Component: abc 4

5. Deployment 4

6. Implementation View 4

Software Architecture Document

# Introduction

## Purpose

This document provides an intensive and comprehensive architecture overview of the software using various architectural views that describe separated components in detail. It depicts different aspects of the software and conveys significant architectural decisions to developers and non-developers.

## Scope

This document provides an intensive and comprehensive architecture overview of the **Courses Roadmap** web application.

## Definitions, Acronyms and Abbreviations

See the Glossary document.

## References

Vision document

Use-case specification document

# Architectural Goals and Constraints

There are many software requirements and constraint that have significant impact on the architecture pointed out as following:

* The Courses Roadmap web application is based on a client-server model. The application runs on all web browsers. The Courses Roadmap web application will be deployed on the dedicated 24/7 server on the Firebase
* Sensitive data must be encrypted
* Every sensitive action must be logged and notified to users via email, including any access on other devices and changing email or password.
* Every non-functional requirement derived in the Vision Document must be considered as the architecture is being developed.

# Use-Case Model

Diagram

Description automatically generated

For the specification, look up the **Use-case specification** document.

# Logical View

Diagram

Description automatically generated

The system includes two large components: Web application (client) and Server. Users use the Web application. Clients and the server communicate through the internet and using the Firebase SDK

On the Web application, we divide into 3-tiers:

* The highest tier is the GUI which represents the view for users to interact and provide the data for the controller.
* The controller tier manipulates the data received from users through the GUI tier and using services tier to access local storage. Moreover, whenever it need to connect with server, it also call the services of API

## Web application architecture

Graphical user interface

Description automatically generated

More detail about the client architecture, the software has the user interface which is represented by the GUI layer including many layouts. The first layer provides services to get the input data from the user and to update the view. The controller layer listens to the user interaction and processes the business logic. The service layer has two main components:

* Cache: to read and write data to local storage
* SDK: to communicate with Firebase Server

## Server architecture

Diagram

Description automatically generated

This project server base on Firebase platform. The first layer is the Authentication Layer, which is used to authorize that the user has permission to access the database and storage resource. This layer uses the user's Google account or email address to verify their identity, after the user has been confirmed, this layer will provide an access token to the web client via Firebase SDK and the user can access to Firebase’s database and storage.

## Class Diagram

Diagram, schematic

Description automatically generated

# Deployment

[Leave this section blank for PA3.]

# Implementation View

[Leave this section blank for PA3.]