

**CAPSTONE PROJECT 2**

**ARCHITECTURE DESIGN DOCUMENT**

**What should I eat today?**

**VERSION: 1.0**

**Mentor : Nguyen Thi Bao Trang**

**Project Team : 101dogS Team**

**Team Member : Le Nguyen Hoang Van**

**Luong Minh Hieu**

**Tran Quang Khai**

**Nguyen Dinh Luu**

**03/22/2020**

**INTERNATIONALSCHOOL OF DUYTAN UNIVERSITY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT INFORMATION** | | | | |
|  | **PROJECT INFORMATION** | | | |
| **Project Acronym** | WIET | | | |
| **Project Title** | What should I eat today? | | | |
| **Start Date** | 02/12/2020 | **Start Date** | 02/12/2020 | |
| **Lead Institution** | International School, Duy Tan University | | | |
| **Project Mentor** | Nguyen Thi Bao Trang | | | |
| **Product Owner & Contact Detail** | Le Nguyen Hoang Van | | | |
| **Partner Organization** |  | | | |
| **Scrum Master** | Le Nguyen Hoang Van | lenguyenhoangvan18@gmail.com | | 0935604934 |
| **Team Members** | Luong Minh Hieu | minhhieu98@gmail.com | | 0399870055 |
| Nguyen Dinh Luu | dinhluu098@gmail.com | | 0935883503 |
| Tran Quang Khai | tquangkhai98@gmail.com | | 0976308098 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ARCHITECTURE DESIGN DOCUMENT** | | | |
| **Document Title** | Architecture Design Document | | |
| **Author(s)** | 101dogS Team | | |
| **Role** |  | | |
| **Date** | 03/22/2020 | **File name:** | ArchitechtureDesign-Caps2-101dogS-ver1.0.pdf |
| **URL** |  | | |
| **Access** | Project and CMU Program | | |

|  |  |  |
| --- | --- | --- |
| **REVISION HISTORY** | | |
| **Version** | **Date** | **Description** |
| 1.0 | 14 - Sep - 2019 | Create Architecture Design Document |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Approval**  The following signatures are required for approval of this document | | | |
| **Mentor** | Nguyen Thi Bao Trang | **Signature:** |  |
| **Date:** |  |
| **Product owner** | Le Nguyen Hoang Van | **Signature:** |  |
| **Date:** |  |
| **Scrum master** | Le Nguyen Hoang Van | **Signature:** |  |
| **Date:** |  |
| **Team member(s)** | Nguyen Dinh Luu | **Signature:** |  |
| **Date:** |  |
| Tran Quang Khai | **Signature:** |  |
| **Date:** |  |
| Luong Minh Hieu | **Signature:** |  |
| **Date:** |  |

**Contents**

[**1.** **Introduction** 5](#_Toc27405614)

[**1.1.** **Purpose** 5](#_Toc27405615)

[**1.2.** **Documents Referenced** 5](#_Toc27405616)

[**2.** **Project Statement** 5](#_Toc27405617)

[**2.1.** **Project Overview** 5](#_Toc27405618)

[**2.2.** **Business Driver** 6](#_Toc27405619)

[**2.2.1.** **Business Problems** 6](#_Toc27405620)

[**2.2.2.** **Business Need** 6](#_Toc27405621)

[**2.3.** **Project Goals** 6](#_Toc27405622)

[**3.** **Architecture Drivers** 7](#_Toc27405623)

[**3.1.** **High-Level Requirements** 7](#_Toc27405624)

[**3.2.** **System Context** 7](#_Toc27405625)

[**3.3.** **Quality Attribute** 7](#_Toc27405626)

[**4.** **Constraints** 8](#_Toc27405627)

[**4.1.** **Business Constraint** 8](#_Toc27405628)

[**4.2.** **Technical Constraint** 8](#_Toc27405629)

[**5.** **High level architecture** 9](#_Toc27405630)

[**5.1.** **Component and Connector view (C&C view)** 9](#_Toc27405631)

[**5.2.** **Module view** 10](#_Toc27405632)

[**5.3.** **Allocation view** 11](#_Toc27405633)

# **Introduction**

# **Purpose**

This specification covers following:

* Brief specification of the project, high level requirement, system context for the system.
* Use case diagram, detail quality attribution.
* Architecture presented by various architecture view types: Component and Connect tor view, Module view, Allocation view.
  1. **Documents Referenced**

|  |  |
| --- | --- |
| **No** | **Reference** |
| **1** | ProductBacklog-Caps2-101dogS-ver1.0.pdf |
| **2** | ProjectPlan-Caps2-101dogS-ver1.1.pdf |

***Table 1: Document Reference***

## **Project Statement**

* 1. **Project Overview**
* Project name: **WHAT SHOULD I EAT TODAY?**
* Development team:

|  |  |
| --- | --- |
| Full name | Position |
| Le Nguyen Hoang Van | Scrum Master, Product Owner |
| Nguyen Dinh Luu | Scrum Member |
| Tran Quang Khai | Scrum Member |
| Luong Minh Hieu | Scrum Member |

***Table 2: Development team***

* 1. **Business Driver**
     1. **Business Problems**
* Customers don’t know what they should eat for breakfast, lunch, also dinner.
* Customers ant to make an app and get data from foody or other apps and generate random dishes for breakfast, lunch, dinner.
* Customers want to implement Machine Learning for this, which type of food users want to use, base on that, we can recommend the correct one.
  + 1. **Business Need**

Customer need an application that can handle the following issues:

* Helps user decide which dishes they want to use by on their interested.
* Helps user decide dishes for breakfast, lunch, also dinner.
* Helps user find the location of the restaurant.
* Helps user ignore allergy ingredient.
* Helps user find vegetarian dishes.
  1. **Project Goals**

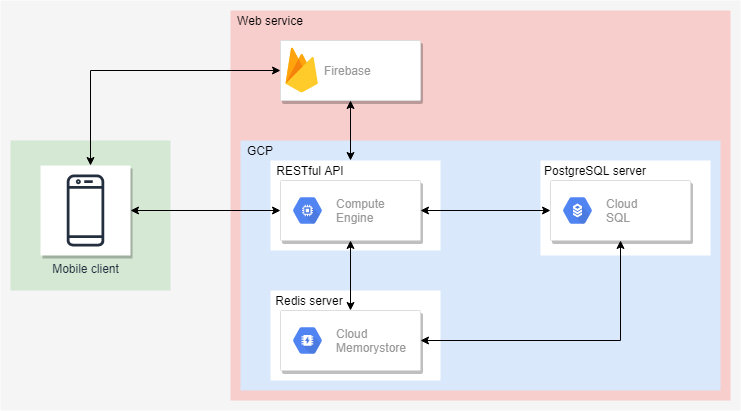
When coach use our application, it can help:

* Reduce time to decide which dishes they want to use.
* Find location easier.
* Personalization of interest in dishes.
* Ignore allergy foods.

1. **Architecture Drivers**
   1. **High-Level Requirements**

Refer to ProductBacklog-Caps2-101dogS-ver1.0.pdf

* 1. **System Context**



***Figure 1: System context***

* **End user:** have full function of system
  1. **Quality Attribute**

|  |  |
| --- | --- |
| **Quality Attribute:** Performance | **ID :** QA01 |
| **Stimulus** | User request function real time |
| **Source(s) of the stimulus** | User |
| **Relevant environmental conditions** | Normal working state of the mobile with Android OS 7.0 or higher |
| **Artifacts** | Application |
| **System response** | The application performance real-time get request and response back on screen |
| **Response measure(s)** | The response is less than 10 seconds |

|  |  |
| --- | --- |
| **Quality Attribute:** Usability | **ID :** QA02 |
| **Stimulus** | Want to feel comfortable and easy to use with the user interface |
| **Source(s) of the stimulus** | User |
| **Relevant environmental conditions** | Normal working state of the mobile with Android OS 7.0 or higher |
| **Artifacts** | Application |
| **System response** | The application provide User-interface easy to use |
| **Response measure(s)** | User no need much time to use this application |

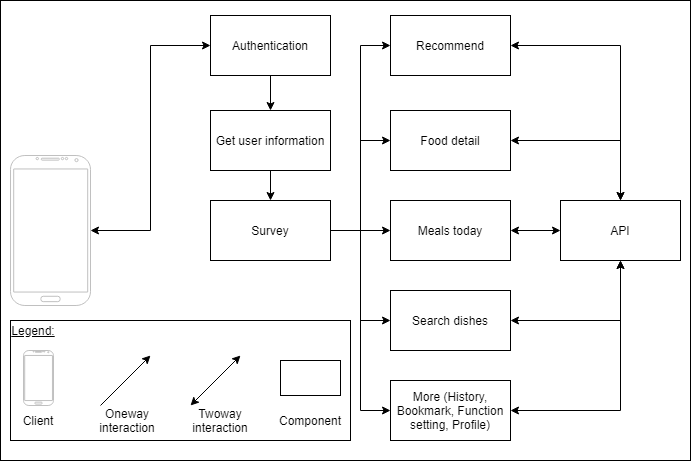
|  |  |
| --- | --- |
| **Quality Attribute:** Security | **ID :** QA03 |
| **Stimulus** | Users want to access to application even through login or not |
| **Source(s) of the stimulus** | User |
| **Relevant environmental conditions** | Normal working state of the mobile with Android OS 7.0 or higher |
| **Artifacts** | Application |
| **System response** | The login user-interface appear |
| **Response measure(s)** | Users need to login to access the application |

1. **Constraints**
   1. **Business Constraint**

* Project will be started on: 02/12/2020.
* Project will be finished on: 05/15/2020.
* Project will be finished in 64 days (960 hours).
  1. **Technical Constraint**
* Technology: Java Android, Python
* Environment:
  + Operating System: Android 7.0 or higher.
  + Develop Tools: Pycharm, Android Studio
  + Source version control: Git (GitLab).
  + Database: PosgreSQL Database.
  + Internet Connection

1. **High level architecture**
   1. **Component and Connector view (C&C view)**

The diagram below shows the overview architecture including component and other related component. We have representations and behaviors for important components in the following sections.



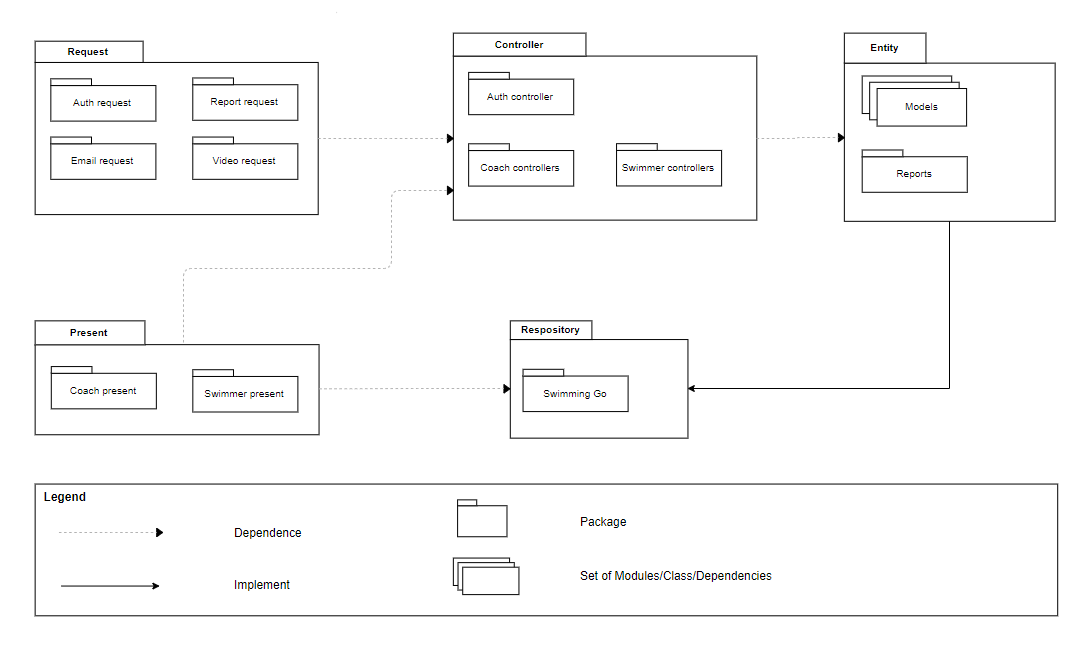
***Figure 2: C&C view***

**Component Mapping:**

* Client: Is the hardware (mobile with Android OS 7.0 or higher)
* Component: is a component that locate on application
* Web service: is a component of Amazon Web Service

**Repository:**

* Database: is a component which contains information of SG system
  1. **Module view**

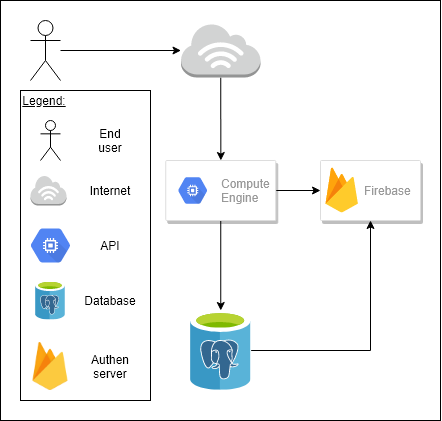
******

***Figure 3: Module view***

**Prose:**

To process and authorize request from user, all packages are tied up together to authorize the association between user and data (Data Authorization Layer). All requests come from users firstly processed by middleware, it will determine user is authenticated to the system or not. Then, controllers can handle the business logic. All data access logic is controlled by repository to handle all the data access logics with the association of criteria layers to filter data of users that they can be retrieved. Finally, depending on endpoints, data will be responded to API Gateway.

* 1. **Allocation view**



***Figure 4: Allocation view***

* **Prose:**

When User access to system via Internet, all requests is processed in a central Web Server. Web Server and Database Servers are associated to the same Security Group, data accessing is only produced between Web Server and Database Servers. Web Server determines action is read or write to allocate request to exact database server to retrieve data or execute insertion request. All write actions are performed in Master Database Server and synchronize to all Slave Database Servers. This architecture will help to reduce overwhelming when read and write actions happened in one central database server.