­­MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Cloud-based Backend as a Service for Building Mobile Applications

|  |  |
| --- | --- |
| Nhóm 7 | |
| Group member | Nguyễn Mạnh Hùng – SE61388 Nguyễn Hữu Lộc – SE61312  Phạm Bảo Toàn – SE61011  Vũ Văn Quyết – SE61071 |
| Supervisor | Kiều Trọng Khánh |
| Ext. Supervisor | N/A |
| Capstone Project code | CSBM |

-Ho Chi Minh, *11-05-2016*-

**Table of Contents**

[**A. Introduction** 5](#_Toc451967954)

[**1.** **Project Information** 5](#_Toc451967955)

[**2.** **Introduction** 5](#_Toc451967956)

[**3.** **Current Situation** 5](#_Toc451967957)

[**4.** **Problem Definition** 5](#_Toc451967958)

[**5.** **Proposed Solution** 6](#_Toc451967959)

[**5.1. Feature functions** 6](#_Toc451967960)

[**5.2. Advantages and Disadvantages** 6](#_Toc451967961)

[**6.** **Functional Requirements** 6](#_Toc451967962)

[**7.** **Role and Responsibility** 7](#_Toc451967963)

[**B. Software Project Management Plan** 8](#_Toc451967964)

[**1.** **Problem Definition** 8](#_Toc451967965)

[**1.1. Name of this Capstone Project** 8](#_Toc451967966)

[**1.2. Problem Abstract** 8](#_Toc451967967)

[**1.3. Project Overview** 8](#_Toc451967968)

[**2.** **Project organization** 11](#_Toc451967969)

[**2.1. Software Process Model** 11](#_Toc451967970)

[**2.2. Roles and responsibilities** 12](#_Toc451967971)

[**2.3. Tools and Techniques** 13](#_Toc451967972)

[**3.** **Project Management Plan** 13](#_Toc451967973)

[**3.1. Product Backlog** 13](#_Toc451967974)

[**3.2. Deliverables** 14](#_Toc451967975)

[**3.3. All Meeting Minutes** 15](#_Toc451967976)

[**4.** **Coding Convention** 15](#_Toc451967977)

[**C. Software Project Management Plan** 16](#_Toc451967978)

[**1.** **User Requirement Specification** 16](#_Toc451967979)

[**1.1. Guest Requirement** 16](#_Toc451967980)

[**1.2. User Requirement** 16](#_Toc451967981)

[**1.3. Authenticated user Requirement** 16](#_Toc451967982)

[**2.** **System Requirement Specification** 16](#_Toc451967983)

[**2.1. External Interface Requirement** 16](#_Toc451967984)

[**2.2.** **System Overview Use Case** 17](#_Toc451967985)

[**2.3. List of Use Case** 18](#_Toc451967986)

[**3.** **Software System Attribute** 27](#_Toc451967987)

[**3.1. Usability** 27](#_Toc451967988)

[**3.2. Reliability** 27](#_Toc451967989)

[**3.3. Availability** 27](#_Toc451967990)

[**3.4. Security** 27](#_Toc451967991)

[**3.5. Maintainability** 27](#_Toc451967992)

[**3.6. Portability** 28](#_Toc451967993)

[**3.7. Performance** 28](#_Toc451967994)

[**4.** **Conceptual Diagram** 28](#_Toc451967995)

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| Name | Definition |
| BaaS | Back-end as a service |

# **A. Introduction**

## **Project Information**

- Project name: **Cloud-based Backend as a Service for Building Mobile**

**Applications.**

- Project code: **CSBM.**

- Project type: **Web Application and Mobile Framework.**

- Start date: **11-05-2016.**

- End date: **21-8-2016.**

## **Introduction**

In this document, we introduce a solution for mobile developers. Developing mobile applications is more and more popular and cloud computing also extremely develops. The combination is very complex to build the effective application on smartphone. Based on our researches and analysis, we proposed a cloud-based backend as a service for mobile developers.

We provide a service that includes website application and mobile framework, which helps mobile developers to develop their application quickly, easily to change, and flexibility in modification. Somethings supported the developer building their application without worrying about the backend services.

## **Current Situation**

When developing mobile applications, developers need to concern too much about back-end such as: creating service support push notification, creating and managing database mobile… Most modern mobile applications store data and interact with other services on the internet. User accounts, shared content, documents and purchases; these things all need to be stored somewhere else, and cloud is a good solution. There is a cloud-based backend as a service of Facebook named “Parse” (https://www.parse.com), however the “Parse” is shutdown in first month 2016…

## **Problem Definition**

Others backend as a service:

* Advantages:
* Parse, Firebase… are well-known and already had much users.
* Disadvantages:
  + Parse (http://parse.com/) is shutdown in first month 2017.
  + CloudKit (<https://developer.apple.com/icloud/>) is limited to iOS and require user to use iCloud.
  + Firebase (<https://www.firebase.com/>) hasn’t push notification

Common mobile development:

* Advantages:
* Taking control everything from mobile layout to back-end component.
* Disadvantages:
* Mobile developers need to have the knowledge about web service or wait provided services from web service developer.
* Mobile developers hard to focus on creating user experience.
* Mobile application development is waste of time on building API.

## **Proposed Solution**

Our proposed solution is to develop the Cloud-based Backend as a Service named “CSBM” to support mobile developer manage application data via our provided framework and only focus on creating extraordinary user experiences. We’ll take care of the rest.

### **5.1. Feature functions**

* Web application: design for developer to manage data and configuration of their mobile application.
* Server: parse data from client side to store and sync data with our NoSQL cloud database.
* Mobile framework: provide framework for mobile developer to interact with our database.

### **5.2. Advantages and Disadvantages**

* **Advantages:**
  + Real-time database: data is stored in cloud database and synced to all connected clients in real-time.
  + Push notification: auto push notification to all connected client after data is changed.
  + Mobile framework for developers to interact with their data on server database.
* **Disadvantages:**
  + Security: users only manage their data while the security is based on the service provided.
  + Application downtime: users can’t control downtime when CSBM server has problem.

## **Functional Requirements**

* **Server component**
  + Parse data from client dashboard to system storage.
  + Notify to connected client.
  + Sync database to server automatically.
* **Web component**
  + Manage applications.
  + Manage data.
* **Mobile framework** 
  + Provide framework for mobile.

## **Role and Responsibility**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Full Name | Role | Position | Contact |
| 1 | Kiều Trọng Khánh | Project Manager | Supervisor | khanhkt@fpt.edu.vn |
| 2 | Nguyễn Mạnh Hùng | Developer | Leader | hungnmse61388@fpt.edu.vn |
| 3 | Nguyễn Hữu Lộc | Developer | Member | locnhse61312@fpt.edu.vn |
| 4 | Vũ Văn Quyết | Developer | Member | quyetvv61071@fpt.edu.vn |
| 5 | Phạm Bảo Toàn | Developer | Member | toanpbse61011@fpt.edu.vn |

*Table 1: Roles and Responsibilities*

# **B. Software Project Management Plan**

## **Problem Definition**

### **1.1. Name of this Capstone Project**

- Official name: Cloud-based Backend as a Service for Building Mobile Applications.

- Vietnamese name: Dịch vụ đám mây hỗ trợ developer xây dựng mobile application.

- Abbreviation: CSBM.

### **1.2. Problem Abstract**

At the moment, especially in Viet Nam, not much developer use BaaS (Backend as a Service). The development of mobile application include building back-end so it requires mobile developer have knowledge about coding web api or they may wait for web api from others developers. This situation makes mobile developers can’t focus on creating good user experiences, and then the effective of build a mobile application is always not best liked developer hope.

Besides, currently there are Parse (<http://parse.com/>), Cloud-Kit (<https://developer.apple.com/icloud/>), Firebase (<https://www.firebase.com/)>,... But they have not provided developers a good way to build a mobile application. Cloud-Kit is limited to iOS and require user to use iCloud. Firebase hasn’t push notification. Parse is stop service provider in first month 2016 and is shutdown in first month 2017, but many applications are there. If Parse shutdown, developers who are using Parse will have difficulty in finding another BaaS like Parse.

For the goal that improving mobile development, we provide a service which base on Parse Server, to reduce the time consuming in mobile development and mobile developers can focus more on user interface and user experiences. By using Parse Server, we develop a website to create new applications, and a service to sharing data between user’s different devices. The service also haves push notifications, if either of those are important to developer applications, our service will likely save develop time.

### **1.3. Project Overview**

#### **1.3.1. Current Situation**

Below are the problems encountered in this project:

* + **Disadvantages:**
    - Parse source code was developed with NodeJs that nobody in team has much knowledge about.
    - Parse source code not public function to allow users create new application.
    - Parse server now can only run one application. If developers have many application they will need many Parse server to run all of them.
    - Team hasn’t much knowledge about how to code and build a mobile framework.
  + **Advantages:**
    - The server is ready to run based on Parse source code.

#### **1.3.2. The Proposed System**

According to technology researches, we found out that Parse server of Facebook can help us resolving the problem of reducing time consuming in mobile development and also help developers who are using Parse now can move to and continue develop application.

To resolve the problem that Parse uses NodeJs as main language for the server, we can use NodeJs Tutorial from <http://www.tutorialspoint.com/nodejs/> for understanding how NodeJs work. We also find the necessary help from NodeJs development forum to solve problems.

We must to know what dashboard send request to server to create a new application to write a function allow developer can create new application from our service dashboard. According to technology researches, we decided AngularJs will be the framework for dashboard.

We assign responsibility in horizontal, it means everyone will do a whole big think and separated from the others. This decision based on that everything about this project is really new for all of the member so let each one focus on one problem is the best way.

##### **1.3.2.1. Website (Dashboard)**

Main web application is a place for developer to create a new application or register account to use main function of the service website. Developer also can use this website to management the data of their application.

* For developer
  + - Create a new application: Developer can create a new application.
    - Manage data: Create/Edit/Remove developer data in application.
  + For guest
    - Register: User can register new account to use the service.

##### **1.3.2.2. Mobile Framework**

This framework is use for Developer’s mobile Application, which provide APIs to control mobile application data. Mobile developers can control their data really easily after download framework which appropriate to their language (Android or iOS) and apply to their application.

#### **1.3.3. Boundaries of the System**

* This service is built basing on Parse server of Facebook. Our main target is improving the mobile development and help developers, whose applications is using Parse now, have an appropriate place to migrate to.
* The complete product includes:
  + A web application that allow:
    - Register account to use main function of the system
    - Create a new application
    - Manage data
  + Mobile Framework that allow:
    - Provide API

#### **1.3.4. Future Plans**

Current service only support for mobile. In future we will:

* + - Provide framework for website developers.

#### **1.3.5. Development Environment**

##### **1.3.5.1. Hardware requirements**

* + For server

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 4 Mbps |
| Operating System | Ubuntu Server 12 LTS | Ubuntu Server 14.04.2 LTS |
| Computer Processor | Intel® CORE i3 Quad core 2.1 GHz | Intel® CORE i7 Quad core 2.4 GHz |
| Computer Memory | 1GB of RAM | 4GB of RAM or more |

*Table 2: Hardware Requirement for server*

* + For website development

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 4 Mbps |
| Operating System | Ubuntu Server 12 LTS | Ubuntu Server 14.04.2 LTS |
| Computer Processor | Intel® CORE i3 Quad core 2.1 GHz | Intel® CORE i7 Quad core 2.4 GHz |
| Computer Memory | 1GB of RAM | 4GB of RAM or more |

*Table 3: Hardware Requirement for website development*

* + For mobile framework development

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirements | Recommended |
| Internet Connection | 512Kbps | 4 Mbps |
| Operating System | Ubuntu Server 12 LTS | Ubuntu Server 14.04.2 LTS |
| Computer Processor | Intel® CORE i3 Quad core 2.1 GHz | Intel® CORE i7 Quad core 2.4 GHz |
| Computer Memory | 1GB of RAM | 4GB of RAM or more |

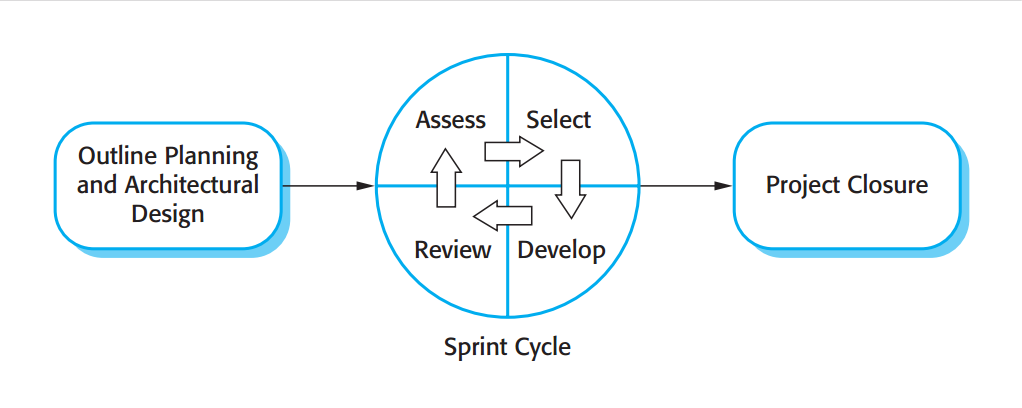
*Table 4: Hardware Requirement for mobile framework development*

## **Project organization**

### **2.1. Software Process Model**

The project is developed under Scrum model. We choose this model because the following reasons:

* + This project is about Parse server, NodeJs and building mobile framework these are really new to all of member. That why our team need focus on research, study and try our best to implement a core flow of this project early.
  + We need to provide many demo for each requirement to make sure that our team can complete this project.
  + This project is about building framework so it’s really hard to finish the document first. We need coding through some demo to understand more about the operation mechanism of Parse server and figure out how to build framework effectively.



*Figure 1: Scrum model (Source: Software Engineering, 9th Edition, Chapter 3, Figure 3.8)*

### **2.2. Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| 1 | Mr. Kiều Trọng Khánh | Scrum Product Owner – Technical Expert | * Specify user requirement * Specifying the business * Control the development process * Give advices on techniques, solutions and business analysis support |
| 2 | Nguyễn Mạnh Hùng | Scrum Master, Business Analyst, Developer, Tester | * Managing process * Researching solutions and techniques * Assigning task for members * Study Parse server for building “Create new app”. |
| 3 | Nguyễn Hữu Lộc | Team Member, Business Analyst, Developer, Tester | * Researching solutions and techniques * Build framework for iOS developers. |
| 4 | Vũ Văn Quyết | Team Member, Business Analyst, Developer, Tester | * Researching solutions and techniques * Build framework for Android developers. |
| 5 | Phạm Bảo Toàn | Team Member, Business Analyst, Developer, Tester | * Researching solutions and techniques * Build Dashboard |

*Table 5: Roles and Responsibilities Details*

### **2.3. Tools and Techniques**

|  |  |
| --- | --- |
| **Tools** | **Name / version** |
| Html, Javascript Editor | Sublime Text V3 |
| iOS IDE | Xcode 7.2 |
| Android IDE | Android Studio 2.1 |
| DBMS | Robomongo 0.9.0 RC8 |
| Source control | GitHub |
| Document | Microsoft Word 2016 |

*Table 6: Tools used for this project*

|  |  |
| --- | --- |
| **Techniques** | **Name / version** |
| Backend | NodeJs 6.2.0 |
| Frontend | AngularJs 1.5.0 |
| iOS Framework | Objective-C 2.0 |
| Android Framework | Java |
| DBMS | MongoDB 3.2.6 |

*Table 7: Techniques used for this project*

## **Project Management Plan**

### **3.1. Product Backlog**

|  |  |  |
| --- | --- | --- |
| **Id** | **User Story** | **Priority** |
| 1 | Create new application | High |
| 2 | Manage applications | High |
| 3 | Manage user account for an application | High |
| 4 | Mange user role for an application | Medium |
| 5 | Multiple application on one Parse server | High |
| 6 | Push notification | Low |
| 7 | Framework for Android | High |
| 8 | Framework for iOS | High |
| 9 | Application search | Low |
| 10 | Login | High |
| 11 | Logout | Low |
| 12 | Auto generate ApplicationId | Medium |
| 13 | Analytics | Low |

### **3.2. Deliverables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverable** | **Delivery date** | **Delivery Location** | **Note** |
| 1 | Introduction Document | 12/5/2016 | FPT-CMS | Report No.1 |
| 2 | Software Project Management Plan | 16/5/2016 | FPT-CMS | Report No.2 |
| 3 | Software Requirement Specification | 30/5/2016 | FPT-CMS | Report No.3 |
| 4 | Application Demo 1 | 16/6/2016 | FPT-University | Core flow demo |
| 5 | Application Demo 2 | 21/7/2016 | FPT-University | Features demo |
| 6 | Design 1 | 21/7/2016 | FPT-CMS | Report No.4 |
| 7 | Application Demo 3 | 7/8/2016 | FPT-University | Semi-final application demo |
| 8 | Design 2 | 7/8/2016 | FPT-CMS | Report No.5 |
| 9 | Completed Document | 14/8/2016 | FPT-CMS | Report No.6 |

### **3.3. All Meeting Minutes**

All meeting minutes here: <https://github.com/hungnm2904/CSBM/tree/report/Meeting%20Minutes>.

## **Coding Convention**

Summary:

* Naming Convention:
  + Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.
  + Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.
  + All names start with a letter.
  + Variable and function names written as **camelCase**.
* Declarations Convention:
  + One declaration per line is recommended since it encourages commenting.

Using Java Code Convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

<http://www.w3schools.com/js/js_conventions.asp>

<https://github.com/mgechev/angularjs-style-guide>

Programming with Objective-C summary:

* Naming Convention:
  + Class names must be unique across an entire app.
  + Method names should be expressive and unique within a Class.
  + Local Variables must be unique within The same scope.
  + Accessor method names Must follow conventions.
  + Object creation method names must follow conventions.

Using Object Convention from:

<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Conventions/Conventions.html#//apple_ref/doc/uid/TP40011210-CH10-SW1>

# **C. Software Project Management Plan**

## **User Requirement Specification**

### **1.1. Guest Requirement**

Guest is a person who does not have access the system. Guest can use some functions in the system. To use all functions, guest must login. These are some functions guest can use:

* Login
* Register

### **1.2. User Requirement**

User is guest who already login to the system by his/her account and uses service of system. The customer can use some following functions:

* Manage application: Create New Application, Delete Application, Active Application.
* Manage class: Create Class, Update Class, Delete Class.
* Query data

### **1.3. Admin Requirement**

Person who manages account. Administrator can use some following functions:

* Change User status.
* Change Application status.

### **1.4. Authorized User Requirement**

Authorized User is the person who has accessed the system, besides the functions that user can use base on their role, authenticated user also can use the following function:

* Logout.

## **System Requirement Specification**

### **2.1. External Interface Requirement**

#### **2.1.1. User interface**

* The user interface use English language in web application and mobile framework.
* The user interface for website display best on 1024x768-screen size.

#### **2.1.2. Hardware interface**

* Computer:
  + OS: Ubuntu 14 LTS
  + CPU: Intel® CORE i3 Quad core 2.1 GHz
  + RAM: 1GB
* Mobile:

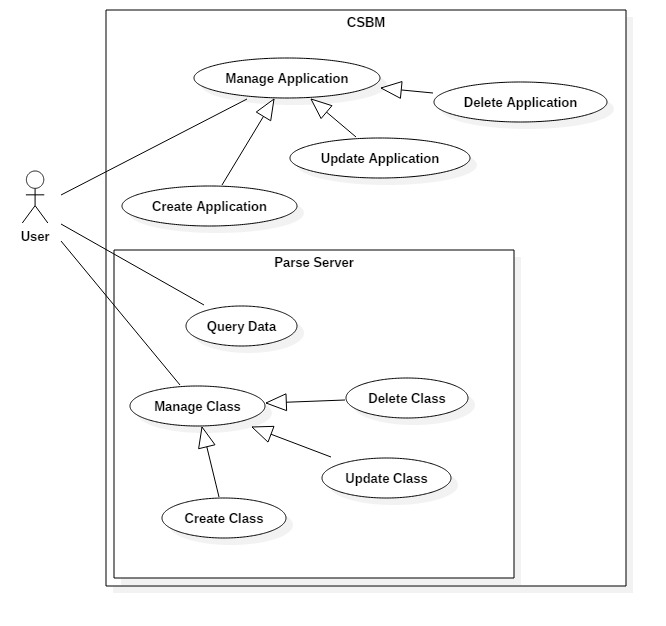
#### **2.1.3. Software Interface**

* Web application: work with Chrome (v47 or above), Internet Explorer (v10 or above), Firefox (v43 or above)
* Mobile framework: Android operation system (v 4.4 or above).

#### **2.1.4. Communication Protocol**

* Use Rest API for communication between the web browser and server.
* Use Rest API for communication between the mobile framework and server.

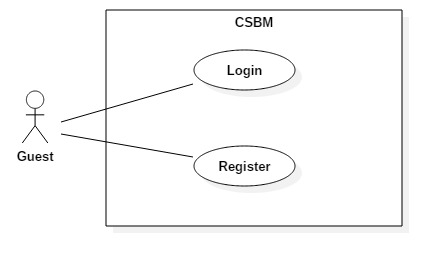
### **System Overview Use Case**



*Figure: System Overview Use Case*

### **2.3. List of Use Case**

#### **2.3.1. <Guest> Overview use case**



*Figure: <Guest> Overview Use Case*

##### **2.3.1.1. <Guest> Register**



*Figure: <Guest> Register*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC001** | | | |
| **Use Case No.** | 001 | **Use Case Version** | 1.0 |
| **Use Case Name** | Register | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * Guest   **Summary:**   * Register a new account.   **Goal:**   * Account is registered successfully and store in database of the system   **Triggers:**   * Guest sends command to register   **Preconditions:** N/A.  **Post Conditions:**   * **Success:** New account will be created * **Fail:** Systemshows messages   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest sends command to request register | System requires information from Guest:   * Email: free text input, required, regex [^[\_A-Za-z0-9-\\+]+(\\.[\_A-Za-z0-9-]+)\*@"+"[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$] * Full name: free text input, required, length (10-50) * Username: free text input, required, length (9-20) * Password: free text input, required, length (6-12) * Repeat password: free text input, required, length (6-12) * Phone: free number input, required, length (10-12) positive integer, value: [0,9] | | 2 | Guest inputs information |  | | 3 | Guest sends command to register  [Alternative 1]  [Alternative 2] | System shows login view  Account registered  [Exception 1]  [Exception 2]  [Exception 3]  [Exception 4] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Guest sends command to reset | System reset all field to blank | | 2 | Guest send command to back to login view | System shows login view  Account isn’t created |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Guest input already exist username | System show warning message “User name already exist” | | 2 | Guest input already exist email | System shows warning message “Email already exist”. | | 3 | Guest does not input required field. | System notices that guest need to input all these field:   * “Email”: System display warning message: “please fill out this field”. * “Full name”: System display warning message: “please fill out this field”. * “Username”: System display warning message: “please fill out this field”. * “Password”: System display warning message: “please fill out this field”. * “Repeat password”: System display warning message: “please fill out this field”. * “Phone”: System display warning message: “please fill out this field”. | | 4 | Guest input wrong some fields with requirement. | System notices that guest need to re-input all these field:   * “Email”: System display warning message: “Email invalid! ([me@example.com)](mailto:me@example.com))”. * “Email”: System display warning message: “Email must be 10 - 254 characters”. * “Full name”: System display warning message: “Full name must be 10 - 50 characters”. * “Username”: System display warning message: “Username must be 6 - 20 characters”. * “Password”: System display warning message: “Password must be 6 - 12 characters”. * “Repeat password”: System display warning message: “Repeat password does not match password”. * “Phone”: System display warning message: “Phone must be numbers”. |   **Relationships:** N/A  **Business Rules:**   * After registered, information of account will be stored in database of the system with role “User” and status is “active” | | | |

**Table: Register specification**

##### **2.3.1.2. <Guest> Login**

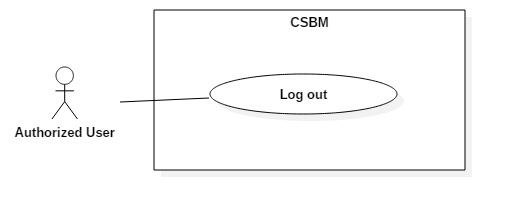


*Figure: <Guest> Login*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC002** | | | |
| **Use Case No.** | 002 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** |  | | |
| **Date** | 25/5/2016 | **Priority** | High |
| **Actor:**   * Guest   **Summary:**   * This use case allows authorized user to log in system.   **Goal:**   * Guest can log in the system.   **Triggers:**   * Guest send the login command.   **Precondition:**   * N/A   **Post Condition:**   * **Success:** User login system. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest goes to login view. | System requires information:   * Username: free text input * Password: free text input | | 2 | User inputs information  [Exception 1,2] |  | | 3 | User sends login command. | Guest will login system with their specified. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | User inputs wrong credential information. | System show error message that wrong username or password. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | User does not input username, or input username including special characters | System notice that user need to input valid username | | 2 | User does not input password, or input password including special characters | System notice that user need to input valid password |   **Relationships:** N/A  **Business Rules:**   * Username: textbox, at least 6 words, required. * Password: textbox, at least 6 words, required. * Logged in users have 2 roles: user and admin. * An access token will be created and one user has only one access token exists. * A cookie will be created and available in 24 hours. | | | |

**Table: Login specification**

#### **2.3.2. <Authorized User> Overview use case**



*Figure: <Authorized User> Overview Use Case*

##### **2.3.2.1. <Authorized User> Logout**

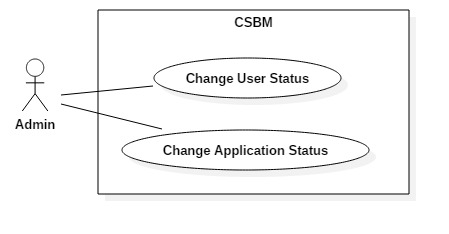


*Figure: <Authorized User> Logout*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC003** | | | |
| **Use Case No.** | 003 | **Use Case Version** | 1.0 |
| **Use Case Name** | Log out | | |
| **Author** |  | | |
| **Date** | 25/5/2016 | **Priority** |  |
| **Actor:**   * Authenticated user.   **Summary:**   * This use case allows Authenticated user logouts the system.   **Goal:**   * User can log out of system.   **Triggers:**   * User sends logout command. * User sends other command after not available too long.   **Precondition:** N/A.  **Post Condition:**   * **Success:** Display login view. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User sends log out command  [Alternative 1] | System clear session state if have, takes user out of system.  System display login view. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor action | System Response | | 1 | User sends any command after not available too long. | System check the session state then takes user out of system.  System display login view. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * After logout, role “Authenticated User” will become “Guest” | | | |

**Table: Logout specification**

#### **2.3.3. <Admin> Overview use case**



*Figure: <Admin> Overview Use Case*

##### **2.3.3.1. <Admin> Change Application Status**

*Figure: <Admin> Change Application Status*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC004** | | | |
| **Use Case No.** | 004 | **Use Case Version** | 1.0 |
| **Use Case Name** | Change Application Status | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * Admin.   **Summary:**   * This use case allows Admin can change Applications status.   **Goal:**   * The system changes the application status.   **Triggers:**   * Admin send the change Application status command.   **Precondition:**   * The Administrator is logged into the system.   **Post Condition:**   * **Success:** Application status is updated into storage. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The Administrator selects a “List Applications” option. | System show a “List Applications” view. | | 2 | The Administrator click toggle button in column Status to Change Application status. | System send the change Application status command. |   **Alternative Scenario:** N/A.  **Exceptions:** N/A.  **Relationships:** N/A.  **Business Rules:**   * If application’s status is changed to “Deactivate”, the application won’t be able to use and users won’t be able to access the application. * In owner view, users won’t be able to access the application has banner “Deactivate” with red text. * If application’s status is changed to “Activate”, users can access the application. | | | |

**Table: Change Application Status specification**

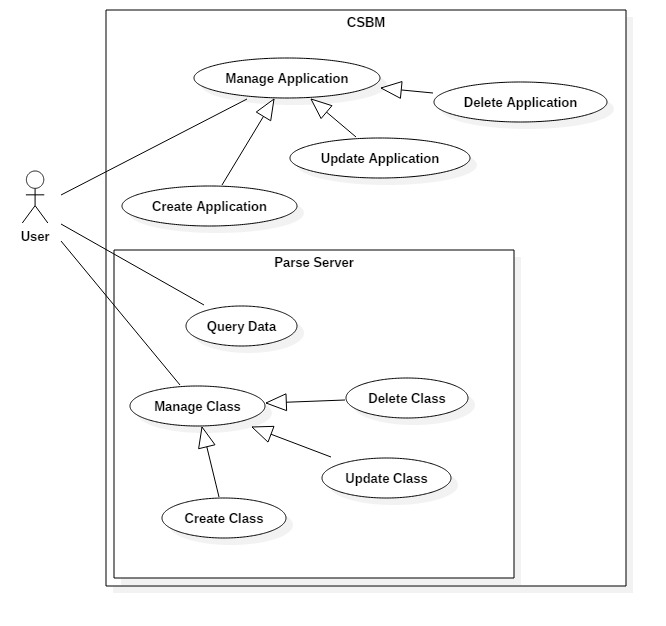
##### **2.3.3.2. <Admin> Change User Account Status**

*Figure: <Admin> Change User Account Status*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC004** | | | |
| **Use Case No.** | 004 | **Use Case Version** | 1.0 |
| **Use Case Name** | Change User Account Status | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * Admin.   **Summary:**   * This use case allows Admin can change User Account status.   **Goal:**   * The system changes the User Account status.   **Triggers:**   * Admin send the change User Account status command.   **Precondition:**   * The Administrator is logged into the system.   **Post Condition:**   * **Success:** The User Account status is change. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | The Administrator selects a “List User Account” option. | System show a “List User Account” view. | | 2 | The Administrator click toggle button in column Status to Change User Account status. | System send the change User Account status command.  [Exception 1] |   **Alternative Scenario:** N/A.  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | The Administrator Deactivate himself. | System show error message. |   **Relationships:** N/A.  **Business Rules:**   * If account’s status is changed to “Deactivate”, the account won’t be able to access the system. * Administrator can’t Deactivate himself. * If account’s status is changed to “Activate”, Guest can use this user account to access the system. | | | |

**Table: Change User Status specification**

#### **2.3.4 <User> Overview use case**



*Figure: <User> Overview Use Case*

##### **2.3.4.1. <User> Create New Application**



*Figure: <User> Create New Application*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC008** | | | |
| **Use Case No.** | 008 | **Use Case Version** |  |
| **Use Case Name** | Create New Application | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * User.   **Summary:**   * This use case allows user to create new application.   **Goal:**   * User creates successfully new application.   **Triggers:**   * User sends the create new application command.   **Precondition:**   * Actor logged in system with role “User”.   **Post Condition:**   * **Success:** New application will be created for user. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User go to “Manage application” and click “Add New Application” button. | System show dialog following information:   * Application Name: textbox, required. * Button “Add New” sends create new application command. | | 2 | User fills in information application. |  | | 3 | User sends command to create new application with inputted information to system (Click “Add New” button).  [Alternative 1] | New application will be created into database.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Staff click “Close” button to cancel process. | System close dialog and cancel create process. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Case | System Response | | 1 | “Application Name” is empty. | Show error message. |   **Relationships:** N/A.  **Business Rules:**   * New Application will be created with status In-Active. * System must ensure has not duplicated application name. * When user add application from create new application form, system will automatic generate userId by Id’s user create application, random ObjectId, masterKey, generate created\_at and updated\_at by current time and save it to system, for example:  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ObjectId | masterKey | created\_at | update\_at | name | userId | | 5759360ae77ed0391e143d45 | 38ccZfmdEJXE5axrKh1TTX7Qp3uAfLVjYZYi6mVQFNPi92hn | ISODate("2016-06-09T09:25:30.000Z") | ISODate("2016-06-09T09:25:30.000Z") | test | 5756aaaa6f9268590fa0867b | | | | |

**Table: Create New Application specification**

##### **2.3.4.2. <User> Delete Application**

*Figure: <User> Delete Application*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC009** | | | |
| **Use Case No.** | 009 | **Use Case Version** |  |
| **Use Case Name** | Delete Application | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * User   **Summary:**   * This use case allows user to delete application in system.   **Goal:**   * User can delete application.   **Triggers:**   * User sends deleting application command.   **Precondition:**   * Actor logged in system with role “User”.   **Post Condition:**   * **Success:** The application is deleted. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User go to “Manage application” and click “Delete Application” button. | System show dialog following information:   * Application: dropdown list application name.   Button “Delete Application” sends delete application command. | | 2 | User choose application name. |  | | 3 | User sends command to delete application to system (Click “Delete Application” button). | System shows message options to confirm command. | | 4 | User confirms to delete application  [Alternative 1] | System delete application in system. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | User cancel confirmation to delete application. | System abort to delete application. |   **Exceptions:** N/A.  **Relationships:** N/A.  **Business Rules:**   * Application cannot be empty. | | | |

**Table: Delete application specification**

##### **2.3.4.3. <User> Update Application**

*Figure: <User> Update Application*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC008** | | | |
| **Use Case No.** | 010 | **Use Case Version** |  |
| **Use Case Name** | Update Application | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * User.   **Summary:**   * This use case allows user to update application.   **Goal:**   * User can update an available application.   **Triggers:**   * User sends updating application command.   **Precondition:**   * Actor logged in system with role “User”. * The application must be available before.   **Post Condition:**   * **Success:** New application data is updated into storage. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User go to “Manage application” and click “Update Application” button. | System show dialog following information:   * Application Name: textbox, required. * Button “Update Application” sends update application command. | | 2 | User fills in information application. |  | | 3 | User sends command to update application with inputted information to system (Click “Update Application” button).  [Alternative 1] | All application data are updated into storage.  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Staff click “Close” button to cancel process. | System close dialog and cancel update process. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Case | System Response | | 1 | “Application Name” is empty. | Show error message. |   **Relationships:** N/A.  **Business Rules:**   * New information of application will be stored in storage. * System must ensure has not duplicated application name. | | | |

**Table: Update Application specification**

##### **2.3.4.4. <User> Update Class Field**

*Figure: <User> Update Class Field*

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC008** | | | |
| **Use Case No.** | 011 | **Use Case Version** |  |
| **Use Case Name** | Change Class Field Name | | |
| **Author** |  | | |
| **Date** | 25/05/2016 | **Priority** |  |
| **Actor:**   * User.   **Summary:**   * This use case allows user to update class field.   **Goal:**   * User can update an available class.   **Triggers:**   * User sends updating class field command.   **Precondition:**   * Actor logged in system with role “User”. * The class must be available before.   **Post Condition:**   * **Success:** New class field data is updated into storage. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User go to “Manage Classes” and click “Update Field” button. | System show dialog following information:   * Field Name: dropdown list, required. * New Field Name: textbox, required. * Button “Update Field” sends update field command. | | 2 | User fills in information field. |  | | 3 | User sends command to update field with inputted information to system (Click “Update Field” button).  [Alternative 1] | All Field data are updated into storage.  [Exception 1,2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Staff click “Close” button to cancel process. | System close dialog and cancel update process. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Case | System Response | | 1 | “Field Name” is empty. | Show error message. | | 2 | “New Field Name” is empty. | Show error message. |   **Relationships:** N/A.  **Business Rules:**   * New information of field will be stored in storage. * System must ensure has not duplicated field name in available class. | | | |

**Table: Update Application specification**

## **Software System Attribute**

### **3.1. Usability**

#### **3.1.1. Graphic user interface**

All the test, labels and alerts of web application and mobile framework will be written by English.

#### **3.1.2. Usability**

The system usability is easy to use that will need less than 1 hour for training to use system.

#### **3.1.3. Installation**

User can follow installation and manual guide for installation. If there are any problems, user cans contacts developer for help.

### **3.2. Reliability**

* The data should be backed up everyday.

### **3.3. Availability**

* The service relates to communication so it can be available 24/7.
* Server should have back-up method to make sure that if it having problems, all necessary data can be protected and restore easily.

### **3.4. Security**

* Input data is validated if necessary before saving to database.
* Users is authenticated/authorized for all users when they logged in to the system.

### **3.5. Maintainability**

* System is separated into modules.

### **3.6. Portability**

* User, guest can use application on every OS supported web browsers.
* User can use mobile framework for develop any Android or iOS application.

### **3.7. Performance**

* Requests from web application are responded in less than 10 seconds at 8 Mbps bandwidth speed.

## **Conceptual Diagram**