|  |
| --- |
|  |

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| Capstone Project Document | |
| **Email marketing platform with Simple Email Service of Amazon Cloud** | |
| Group 3 -JS | |
| **Group members** | Ngô Minh Tấn – SE62503 (Leader)  Nguyễn Lưu Hồng Sơn – SE62414  ~~Đặng Võ Anh Khoa – SE61550~~  Nguyễn Hữu Thắng –SE62447 |
| **Supervisor** | Lâm Hữu Khánh Phương |
| **Ext. Supervisor** | N/A |
| **Project Code** | EMP |

**– Ho Chi Minh City, May 13th, 2019 –**

This page is intentionally left blank

**Table of Contents**

[**Table of Contents** 3](#_Toc7065699)

[A. Introduction 12](#_Toc7065700)

[1. Project Information 12](#_Toc7065701)

[2. Introduction 12](#_Toc7065702)

[3. Current Situation 12](#_Toc7065703)

[4. Problem Definition 13](#_Toc7065704)

[5. Proposed Solution 13](#_Toc7065705)

[5.1. Functions 13](#_Toc7065706)

[5.2. Advantages and disadvantages 14](#_Toc7065707)

[6. Functional Requirements 15](#_Toc7065708)

[6.1. Registered User website and mobile application 15](#_Toc7065709)

[6.2. Admin website 15](#_Toc7065710)

[7. Role & Responsibility 16](#_Toc7065711)

[B. Software Project Management Plan 17](#_Toc7065712)

[1. Problem Definition 17](#_Toc7065713)

[1.1. Name of this Capstone Project 17](#_Toc7065714)

[1.2. Problem Abstract 17](#_Toc7065715)

[1.3. Project Overview 17](#_Toc7065716)

[2. Project Organization 21](#_Toc7065717)

[2.1. Software Process Model 21](#_Toc7065718)

[2.2. Roles and responsibilities 21](#_Toc7065719)

[2.3. Tools and Techniques 23](#_Toc7065720)

[3. Project Management Plan 23](#_Toc7065721)

[3.1. Product Backlog 23](#_Toc7065722)

[3.2. Sprint Backlog 23](#_Toc7065723)

[3.3. Deliverables 23](#_Toc7065724)

[3.4. All Meeting Minutes 24](#_Toc7065725)

[4. Coding Convention 24](#_Toc7065726)

[C. Software Requirement Specification 25](#_Toc7065727)

[1. User Requirement Specification 25](#_Toc7065728)

[1.1. Guest requirement 25](#_Toc7065729)

[1.2. Registered User requirement 25](#_Toc7065730)

[1.3. Staff requirement 26](#_Toc7065731)

[1.4. Admin requirement 27](#_Toc7065732)

[2. System Requirement Specification 27](#_Toc7065733)

[2.1. External Interface Requirement 27](#_Toc7065734)

[2.2. System Overview Use Case 29](#_Toc7065735)

[2.3. List of use case 31](#_Toc7065736)

[3. Software System Attribute 205](#_Toc7065737)

[3.1. Usability 205](#_Toc7065738)

[3.2. Reliability 205](#_Toc7065739)

[3.3. Availability 206](#_Toc7065740)

[3.4. Security 206](#_Toc7065741)

[3.5. Maintainability 206](#_Toc7065742)

[3.6. Portability 206](#_Toc7065743)

[3.7. Performance 206](#_Toc7065744)

[4. Conceptual Diagram 207](#_Toc7065745)

[D. Software Design Description 210](#_Toc7065746)

[1. Design Overview 210](#_Toc7065747)

[2. System Architecture Design 211](#_Toc7065748)

[2.1. Admin web application: 212](#_Toc7065749)

[2.2. Registered User web system: 212](#_Toc7065750)

[3. Component Diagram 213](#_Toc7065751)

[4. Detailed Description 215](#_Toc7065752)

[4.1. Class Diagram 215](#_Toc7065753)

[4.2. Class Diagram Explanation 217](#_Toc7065754)

[4.3. Interaction Diagram 239](#_Toc7065755)

[5. Interface 257](#_Toc7065756)

[5.1. Component Interface 257](#_Toc7065757)

[5.2. User Interface Design 257](#_Toc7065758)

[6. Database Design 307](#_Toc7065759)

[6.1. Entity Relationship Diagram (ERD) 307](#_Toc7065760)

[6.2. Data Dictionary 309](#_Toc7065761)

[7. AlgorithEMP 329](#_Toc7065762)

[7.1. Price suggestion model 329](#_Toc7065763)

[E. System implementation & Test 335](#_Toc7065764)

[1. Introduction 335](#_Toc7065765)

[1.1. Overview 335](#_Toc7065766)

[1.2. Test Approach 335](#_Toc7065767)

[2. Database Relationship Diagram 335](#_Toc7065768)

[2.1. Physical Diagram 335](#_Toc7065769)

[2.2. Data dictionary 337](#_Toc7065770)

[3. Test Plan 339](#_Toc7065771)

[3.1. Features to be tested 339](#_Toc7065772)

[3.2. Features not to be tested 339](#_Toc7065773)

[4. System Testing Test Case 340](#_Toc7065774)

[4.1. Registered User testing case: 340](#_Toc7065775)

[F. Software User’s Manual 343](#_Toc7065776)

[1. Installation Guide 343](#_Toc7065777)

[1.1. Setting up environment at server 343](#_Toc7065778)

[1.2. Deployment at server side 344](#_Toc7065779)

[1.3. Deploy web application on server 345](#_Toc7065780)

[2. User Guide 346](#_Toc7065781)

[2.1. Web application 346](#_Toc7065782)

[2.2. Mobile application 346](#_Toc7065783)

[G. Appendix 357](#_Toc7065784)

**List of tables**

[Table 1:Roles and Responsibilities 15](#_Toc16974818)

[Table 2: Hardware Requirements for Server For PC 18](#_Toc16974819)

[Table 3:Hardware requirement for PC 18](#_Toc16974820)

[Table 4:Development Environment - Software Requirement 19](#_Toc16974821)

[Table 5: Roles and responsibilities 21](#_Toc16974822)

[Table 6: Tools and techniques 21](#_Toc16974823)

[Table 7: Deliverables 22](#_Toc16974824)

[Table 8:<Guest> Login 29](#_Toc16974825)

[Table 9:<Guest> Register 31](#_Toc16974826)

[Table 10:<Registered User> Log out 34](#_Toc16974827)

[Table 11:<Registered User> View profile 36](#_Toc16974828)

[Table 12:<Registered User> Edit profile 37](#_Toc16974829)

[Table 13:<Registered User> View Report 39](#_Toc16974830)

[Table 14<Registered User> View contact 41](#_Toc16974831)

[Table 15:<Registered User> Add Contacts 43](#_Toc16974832)

[Table 16<Registered User> Delete contacts 45](#_Toc16974833)

[Table 17<Registered User> Search contact 46](#_Toc16974834)

[Table 18<Registered User> View templates 48](#_Toc16974835)

[Table 19<Registered User> Create templates 49](#_Toc16974836)

[Table 20<Registered User> Edit template 51](#_Toc16974837)

[Table 21<Registered User> Duplicate Template 52](#_Toc16974838)

[Table 22<Registered User> View group 54](#_Toc16974839)

[Table 23<Registered User> Edit group 55](#_Toc16974840)

[Table 24<Registered User> Create group 57](#_Toc16974841)

[Table 25<Registered User> Search Group 59](#_Toc16974842)

[Table 26<Registered User> Delete Group 60](#_Toc16974843)

[Table 27<Registered User> View appointment 62](#_Toc16974844)

[Table 28<Registered User> Create Appointment 64](#_Toc16974845)

[Table 29<Registered User> View Embed form 66](#_Toc16974846)

[Table 30<Registered User> Create Embed form 67](#_Toc16974847)

[Table 31<Registered User> View Campaigns 69](#_Toc16974848)

[Table 32<Registered User> Create Regular Campaigns 72](#_Toc16974849)

[Table 33<Registered User> Edit Campaigns 73](#_Toc16974850)

[Table 34<Registered User> View workflow 75](#_Toc16974851)

[Table 35<Registered User> Create workflow 77](#_Toc16974852)

[Table 36<Registered User> Create segment 79](#_Toc16974853)

[Table 37<Admin> Login to web admin 81](#_Toc16974854)

[Table 38<Admin> Login to web admin 82](#_Toc16974855)

[Table 39<Registered User> Add mindsending templates 84](#_Toc16974856)

[Table 40 Conceptual Diagram Dictionary 87](#_Toc16974857)

[Table 41 Component Data Dictionary 92](#_Toc16974858)

[Table 42 Class Diagram dictionary 94](#_Toc16974859)

[Table 43- <Class Diagram> Account Attributes 94](#_Toc16974860)

[Table 44- <Class Diagram> Account Methods 94](#_Toc16974861)

[Table 45- <Class Diagram> Role Attributes 95](#_Toc16974862)

[Table 46- <Class Diagram> Role Methods 95](#_Toc16974863)

[Table 47- <Class Diagram> Group Attributes 96](#_Toc16974864)

[Table 48- <Class Diagram> Group Methods 96](#_Toc16974865)

[Table 49- <Class Diagram>Contact Atributes 97](#_Toc16974866)

[Table 50- <Class Diagram>Contact Methods 97](#_Toc16974867)

[Table 51- <Class Diagram>Group contact has contact Attributes 98](#_Toc16974868)

[Table 52- <Class Diagram> Group contact has contact Methods 98](#_Toc16974869)

[Table 53 - <Class Diagram>Campaign Attributes 99](#_Toc16974870)

[Table 54- <Class Diagram> Campaign Methods 99](#_Toc16974871)

[Table 55- <Class Diagram>Campaign has group contact Attributes 100](#_Toc16974872)

[Table 56- <Class Diagram> Campaign has group contact Methods 100](#_Toc16974873)

[Table 57- <Class Diagram>Appointment Attributes 101](#_Toc16974874)

[Table 58- <Class Diagram> Appointment Methods 101](#_Toc16974875)

[Table 59- <Class Diagram>Appointment has group contact Attributes 102](#_Toc16974876)

[Table 60- <Class Diagram> Appointment has group contact Methods 102](#_Toc16974877)

[Table 61- <Class Diagram>Embedded Form Attributes 103](#_Toc16974878)

[Table 62- <Class Diagram> Embedded Form Methods 103](#_Toc16974879)

[Table 63- <Class Diagram>Template Attributes 104](#_Toc16974880)

[Table 64- <Class Diagram> Template Methods 104](#_Toc16974881)

[Table 65- <Class Diagram>Workflow Attributes 105](#_Toc16974882)

[Table 66- <Class Diagram> Workflow Methods 105](#_Toc16974883)

[Table 67- <Class Diagram>Task Attributes 105](#_Toc16974884)

[Table 68- <Class Diagram> Task Methods 106](#_Toc16974885)

[Table 69 <Fields> Login 117](#_Toc16974886)

[Table 70 <Buttons/Hyperlinks> Login 118](#_Toc16974887)

[Table 71 <Fields> Sign Up 119](#_Toc16974888)

[Table 72 <Buttons/Hyperlinks> Sign Up 120](#_Toc16974889)

[Table 73 <Buttons/Hyperlinks> Main menu 122](#_Toc16974890)

[Table 74 <Buttons/Hyperlinks> Campaign Menu 123](#_Toc16974891)

[Table 75 <Buttons/Hyperlinks> Group 123](#_Toc16974892)

[Table 76 <Buttons/Hyperlinks> Embed form 124](#_Toc16974893)

[Table 77 <Buttons/Hyperlinks> appointment 124](#_Toc16974894)

[Table 78 <Fields> Equipment search 125](#_Toc16974895)

[Table 79 <Buttons/Hyperlinks> template 126](#_Toc16974896)

[Table 80 Entity Relationship Diagram Data dictionary 127](#_Toc16974897)

[Table 81 Physical diagram data dictionary 135](#_Toc16974898)

[Table 82 Physical data dictionary 137](#_Toc16974899)

[Table 83 <Test Case> Create new regular campaign 140](#_Toc16974900)

[Table 84 <Test Case> Create new workflow 141](#_Toc16974901)

[Table 85 <Test Case> Create new group 141](#_Toc16974902)

[Table 86 <Test Case> Create new embedded form 142](#_Toc16974903)

[Table 87 <Test Case> Create new appointment 146](#_Toc16974904)

[Table 88 <Test Case> Create new contact 148](#_Toc16974905)

[Table 89 <Test Case> Create new template 149](#_Toc16974906)

[Table 90 <Test Case> Create profile 149](#_Toc16974907)

[Table 91 <Test Case> 新しい通常のキャンペーンを作成する 152](#_Toc16974908)

[Table 92 <Test Case> 新しいグループを作成する 153](#_Toc16974909)

[Table 93 <Test Case> 新しい自動化キャンペーンを作成する 154](#_Toc16974910)

[Table 94 <Test Case> 新しい埋め込みフォームの作成 156](#_Toc16974911)

[Table 95 <Test Case> 新しい予定の作成 160](#_Toc16974912)

[Table 96 <Test Case> 新しい連絡先の作成 164](#_Toc16974913)

[Table 97 <Test Case>新しいテンプレートの作成 165](#_Toc16974914)

[Table 98 <Test Case>新しいテンプレートの作成 166](#_Toc16974915)

[Table 99 - Hardware requirements for Server 167](#_Toc16974916)

[Table 100- Software requirements for Server 167](#_Toc16974917)

[Table 101 - Software requirements for Server 172](#_Toc16974918)

**List of Figures**

[Figure 1 - Scrum Framework 23](#_Toc7065932)

[Figure 2: <Guest> Overview use case 33](#_Toc7065933)

[Figure 3 <Guest> Login 33](#_Toc7065934)

[Figure 4 Conceptual Diagram 209](#_Toc7065935)

[Figure 5 System Architecture diagram 213](#_Toc7065936)

[Figure 6 Component Diagram - Registered User Web Api 215](#_Toc7065937)

[Figure 7 Component Data Dictionary 217](#_Toc7065938)

[Figure 8 Class Diagram 218](#_Toc7065939)

[Figure 9 Class Diagram Dictionary 222](#_Toc7065940)

[Figure 10 Sequence Diagram - Search equipment 242](https://d.docs.live.net/fe447eae7a2d9c8b/Capstone/reports/Report-full.docx#_Toc7065941)

[Figure 11 Sequence Diagram - Login 243](https://d.docs.live.net/fe447eae7a2d9c8b/Capstone/reports/Report-full.docx#_Toc7065942)

[Figure 12 Sequence Diagram - Check Subscription 244](https://d.docs.live.net/fe447eae7a2d9c8b/Capstone/reports/Report-full.docx#_Toc7065943)

[Figure 13 Sequence Diagram - Post equipment 245](#_Toc7065944)

[Figure 14 Sequence Diagram - Request equipment 246](#_Toc7065945)

[Figure 15 Sequence Diagram - Chang hiring status 247](https://d.docs.live.net/fe447eae7a2d9c8b/Capstone/reports/Report-full.docx#_Toc7065946)

[Figure 16 Sequence Diagram - Checkout Materials 248](#_Toc7065947)

[Figure 17 Sequence Diagram - Publish Debris Post 249](#_Toc7065948)

[Figure 18 Sequence Diagram - Bid on Debris Post 250](#_Toc7065949)

[Figure 19 Sequence diagram - Verify Registered User 255](#_Toc7065950)

[Figure 20 Sequence diagram - Add staff 256](https://d.docs.live.net/fe447eae7a2d9c8b/Capstone/reports/Report-full.docx#_Toc7065951)

[Figure 21 Sequence diagram - Add equipmen type 257](#_Toc7065952)

[Figure 22 Edit equipment type 258](https://d.docs.live.net/fe447eae7a2d9c8b/Capstone/reports/Report-full.docx#_Toc7065953)

[Figure 23 Login 259](#_Toc7065954)

[Figure 24 Entity Relation Diagram 310](#_Toc7065955)

[**Figure 25 Linear regression plot - from Wikipedia** 332](#_Toc7065956)

[**Figure 26 Gradient Descent Plot** 334](#_Toc7065957)

[Figure 27: Physical Diagram 338](#_Toc7065958)

[Figure 28: Data Dictionary 341](#_Toc7065959)

# Introduction

## Project Information

* Project name:  **Email marketing platform with Simple Email Service of Amazon Cloud**
* Project Code: **EMP**
* Product Type: **Website Application**
* Start Date: **May 13th, 2019**
* End Date: **July 24th, 2019**

## Introduction

This document introduce a solution for email marketing. At its most basic level, email marketing is the use of email to promote your business. It is used to cultivate relationships with potential customers, keep current customers informed and updated on your brand, offer coupons to encourage customer loyalty, and more.

It is a direct form of marketing, similar to marketing through snail mail, but email marketing is much more efficient for your wallet and for the environment since it’s totally paperless.

Email marketing can be used to build trust with customers over time to turn them into repeat customers. It is also an effective way to keep your customers informed about new sales or promotions you are running. People want to stay informed about your brand, and email marketing is one of the best ways to stay engaged with that audience.

Because of that we have built a software called Mindsending. Unlike others email marketing applications on the market, we not only bring every main function of the email marketing application but also help the user by supporting them the drawing of BPMN (Business Process Modeling Notation) which can be applied to the system so that they can manage and track their work flow in the easiest way.

Beside that our web system uses Amazon SES which will help you for sending a large amount of email from sender to many recipients without sending spam mails, porn mails or in-appropriate content.

## Current Situation

- Email marketing is the act of sending a commercial message, typically to a group of people, using email. Email marketing is significantly cheaper and faster than traditional mail, mainly because with email, most of the cost falls on the recipient. Email marketing carries the benefit of allowing marketers to identify returns on investment and measure and improve efficiency.

- But email marketing also has some of disadvantages: spam emails, porn emails, in-appropriate content in the message which send to the customers.

## Problem Definition

Below are disadvantages of current situation:

* **Traditional marketing costs are very high,** sales companies spend a lot of money on 30s advertising on television or on other media such as newspapers and flyers.
* **Sending traditional emails is easy to send your mail to the spam mailbox,** making the approach to customers more restrictive.
* **Others email marketing applications on the market, it is only for users to create a simple pure automated email** with a non-general look that will make it difficult for new entrants to set up an email marketing.

## Proposed Solution

Our proposed solution is to build a system called MindSending (EMP) to resolve the current situation by helping people has a more general view of the system to automatically send emails to customers by supporting the drawing of BPMN (Business Process Modeling Notation) and then apply to the system so that they can manage and track their work flow in the convenient way.

The EMP system includes two web applications with BPMN drawing tool.

### Functions

* Administrator system (Web application):
  + **Manage Report**: Admin or staff can manage, adjust information with high authority actions.
  + **Manage account**: Admin or staff can check and verify sensitive information, account profile information.
  + **Manage sever**: Admin can view overall sever and manage sever with high authority actions.
* Client applications (Web application):
  + **Manage own account:** The customer has an overview of the account information….
  + **Manage report:** The customer can view report generated from the system by date, month, year.
  + **Manage contact:** When a contact is brought into the system, the customer can add, delete, edit contact and can export contact lists.
  + **Manage group:** Customers can categorize contacts into groups to facilitate contact entry into the appropriate sending email list.
  + **Manage templates:** Customers can add a new template or edit an existing template of the system.
  + **Manage campaign:** Customers can add a new campaign or edit an existing campaign of the system.
  + **Draw BPNM model:** Customers can draw a BPNM model to set up automation campaign.

### Advantages and disadvantages

The advantages and disadvantages of the proposed solution:

* Advantages
  + **Easy for managing and review:** Managing all workflow automation email marketing through BPMN model. Customer can drag/drop object from BPMN drawing tool.
  + **Helping for using contact database effectively through contact and group system EMP:** Help customers categorize customers according to each status.
* Disadvantages:
  + **Can not satisfy all demands:** Marketing is a very flexible and complicated field and there are special business cases that the system cannot fit. Staff involvement may be required in order to resolve these special situations.

## Functional Requirements

Function requirements of the system are listed as below:

### Email marketing website

* Customer can login
* Customer can logout
* Customer can register
* Customer can view account information
* Customer can edit account information
* Customer can search for contact with filter and sorting
* Customer can manage contact in the system
* Customer can manage and change their contact status
* Customer can search for group contact with filter and sorting
* Customer can manage group contact
* Customer can import contact
* Customer can edit contact
* Customer can add new template
* Customer can edit templates suitable for work purposes
* Customer can add new BPNM model
* Customer can edit BPNM model
* Customer can view analysis report by date, month, year

### Admin website

* Admin can manage report
* Admin can manage all accounts on the system
* Admin can manage sever

## Role & Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Lâm Hữu Khánh Phương | Project Manager | Supervisor |  |
| 2 | Ngô Minh Tấn | Developer | Leader |  |
| 3 | Nguyễn Lưu Hồng Sơn | Developer | Member |  |
| 4 | ~~Đặng Võ Anh Khoa~~ | ~~Developer~~ | ~~Member~~ |  |
| 5 | Nguyễn Hữu Thắng | Developer | Member |  |

Table :Roles and Responsibilities

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* Official name: Email marketing platform with Simple Email Service of Amazon Cloud
* Vietnamese name: Ứng dụng khởi tạo email tiếp thị kết hợp dịch vụ email của Amazon
* Abbreviation: EMP

### Problem Abstract

This project concentrates on creating an automation email marketing. It is called Mindsending (EMP). EMP is to assist customer for managing and initiating an automation email marketing.

In marketing, planning a marketing plan that suits the needs of customers is a central issue. Use email that has been collected effectively and can be reused.

The Mindsending web application supports customer an optimal solution for that problem, providing a BPNM model for easy manage and create a workflow.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* **Limited time and human resource.** The team only has 4 members and 13 weeks to execute the project from requirement analytics, design, implement and testing.
* **Lack of business view.** Team members’ major is Software engineering, leading to little experience in general business fields, especially in the construction business field. Therefore, research and training effort would be made to adjust and improve the business model.
* **Lack of experience**. Team members do not have much experience in applying new technology into real projects. There would be new probleEMP arise that require research time and expertise.
* **Schedule conflict.** Team members have different schedule arrangement and lack of co-working time, leading to lack of synchronization of the team.
* **Lack of UX/UI skills.** As developers, the team do not have much knowledge in graphics design and user experience designing, which is a critical factor for a successful C2C product (customer-to-customer product)

#### The Proposed System

According to the recent trend in e-commerce, email marketing is a promising field that can solve access customer’s problem. Therefore, applying that model with necessary modification would be a possible solution.

The basic idea is that we create an email marketing platform with Simple Email Service of Amazon Cloud. We provide a website for end user (customer) to use the system. We also have an admin website to control the system with high authority.

#### Boundaries of the System

* The language of the system is English
* (pending)

#### Future Plans

* Current system can only support admin to manage equipment. We design the system to provide analytic views for admin:
  + Admin can view statistics and visualized graph about transaction process
  + Admin can view strategic information about equipment trends, hiring trends, Registered User needs trends
* Current system can only suggest price to Registered User based on equipment type. We design the system to support data for further machine learning features:
  + Suggest equipment price based on equipment name, location, time
* Current system can only provide delivery status to delivery management. We design the system to support more feature to delivery phase:
  + Support direction and route guidance for delivery
  + Support delivery time estimate based on city, time, traffic

#### Development environment

##### Hardware requirements

For server

|  |  |  |
| --- | --- | --- |
| Hardware | Minimum Requirement | Recommended |
| Internet Connection | Cable, Wi-Fi (8 Mbps) | Cable, Wi-Fi (50 Mbps or more) |
| Operation System | XP, Vista, 7, 10, Window Server 2008, Linux | 10, Window server 2008 |
| Computer Processor | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| Computer memory | 4GB RAM | 32 GB RAM or more |
| Storage space | 1GB | 5GB or more |

Table : Hardware Requirements for Server For PC

|  |  |  |
| --- | --- | --- |
| PC | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window 7 | Window 7 or more. |
| Computer Processor | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| Computer Memory | 1GB RAM | 2GB RAM or more |
| Web Browser | Chromes (v42 or higher) | Chrome latest stable version |

Table :Hardware requirement for PC

##### Software requirement

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operation System | Windows Server 2014 | Operating system and platform for development |
| Environment | Java EE 8.0, Node v10, npm v6 | Specification for developing web application |
| Modeling tool | StarUML, Draw.io | Used to design diagram |
| IDE | Intellij IDEA 2019.1.5, Visual Studio Code 1.27.2 | Programming tools |
| DBEMP | MYSQL 8.0 | Used to create & manage the database for system |
| Source control | Git (Github server) | Used for source control |
| Web browser | Chrome 69 or above | Testing browser |

Table :Development Environment - Software Requirement

## Project Organization

### Software Process Model

This project is developed using Scrum model – part of an agile framework for Software development project. Scrum model is chosen because of the following reasons:

* Scrum is the most suitable model for small and medium project, and we only have 4 members
* The term “machine learning” is new to the team and there would be many trial-and-error tests with the system design before applying to the official release
* There are many new technologies that team members have to learn and develop at the same time, so SCRUM would allow us to do that
* The business model may not be fully tested in real market condition so changes are highly possible. Using Scrum team members can adapt to changes better



Figure 1 - Scrum Framework

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Ngô Minh Tấn | Product Owner | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| 2 | Nguyễn Hữu Thắng | Scrum Master | * Managing project * Designing databases * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing * Arrange Meeting * Risk Management |
| 3 | Nguyễn Lưu Hồng Sơn | Scrum team member | * Clarifying requirements * Preparing documents * Mobile GUI Design * Coding * Testing |

Table : Roles and responsibilities

### Tools and Techniques

|  |  |
| --- | --- |
| Tool/Technique | Name |
| Front-end | HTML, CSS, JavaScript, jQuery, ReactJs, React Native |
| Back-end | JavaEE, Spring, JPA, Hibernate |
| IDE | IntelliJ IDEA 17.2, Visual Studio Code |
| DBEMP | MySQL |
| Source Control | Git, Github |
| Modelling tool | StarUML, Draw.io |

Table : Tools and techniques

## Project Management Plan

### Product Backlog

Product Backlog could be found [here.](https://drive.google.com/open?id=1hJ2loRGJDPmh-D2Zm1oSJTu9u1qBt99ybJAj1JLJSt0)

### Sprint Backlog

Sprint Backlog can be found [here](https://docs.google.com/spreadsheets/d/1ei8C6OGqnKYWK0I5HX1T0Bcu-IQfVVulU_9_QmHVhEY/edit?usp=sharing)

### Deliverables

|  |  |  |
| --- | --- | --- |
| No | Deliverable | Note |
| 1 | Introduction, Entity Relationship Diagram, Business model detail, | Sprint 1 |
| 2 | Use case Overview, UI for mobile application, Conceptual Diagram, First deployment | Sprint 2 |
| 3 | Class Diagram, Physical Diagram, UI for Website Admin, Implement Search equipment | Sprint 3 |
| … |  |  |

Table : Deliverables

### All Meeting Minutes

All sprint meeting minutes could be found [here](https://drive.google.com/open?id=1LIwLFeSkVkFuURBqeYgy2I-vJnobL-u3).

## 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.
* Indentation:
  + One declaration per line is recommended since it encourages commenting.
  + In absolutely no case should variables and functions be declared on the same line.
  + Do not put different types on the same line.
* 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>

# Software Requirement Specification

## User Requirement Specification

### Guest Requirement

***Guest*** *is a person who doesn’t have access to the system. Guest can use some*

*functions in the system. To use all functions, guest must login. These are some*

*functions that guest can use*:

* Login.
* Register.

### User requirement

***User*** *is a guest who logged into the system with user’s role. There are some functions that user can use:*

* Manage own account
* Manage report:

-View report

-Analysis report

* Manage contact:

-Add contact

-Delete contact

-Edit information of contact

-Import list contact

* Manage group:

-Add contact to group

-Delete group

-Edit group

-Search group

-Create group

- Segment contact to group

* Manage templates:

-Add formatted template

-Edit inside template

* Manage campaign:

-Add campaign

-Edit campaign

-Delete campaign

* Draw BPNM model:

-Add BPNM

-Edit BPNM

-Delete BPNM

* View embed form

-Copy script

* Manage invite mail

### Admin requirement

***Admin*** *is an authorized user has permission to log in to the system. An admin has higher authority than user and can do whatever a user can. In addition, an admin can do the following functions:*

* Admin can manage account
* Admin can manage report
* Admin can manage sever

## System Requirement Specification

### External Interface Requirement

#### User Interface:

* GUI should be simple, clear, intuitive, and reminiscent.
* The interface design is an iterate process includes: design, sketching, prototyping, user assessment.
* The language for User Interface is English

#### Hardware Interface

* Mobile Device
* Desktop PC or Laptop

#### Software Interface

* Web application: work with browsers Firefox (v52 or higher), Chromes (v28 or higher), Internet Explorer (v10 or above) or with any web browser that supports HTML5 & CSS3.
* Mobile application: iOS (9.0 or higher)

#### Communication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server.

### 2.2. System Overview Use Case

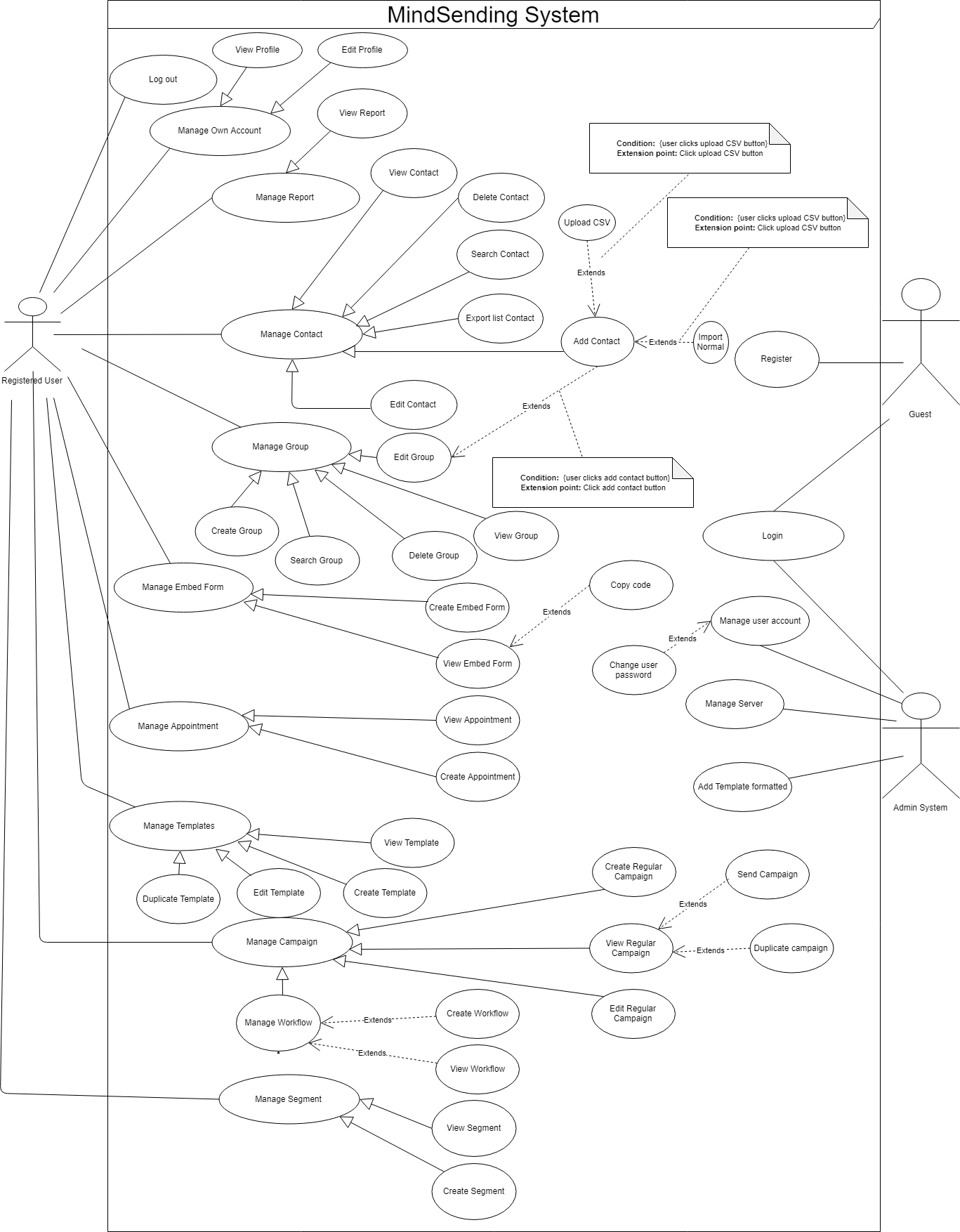


Figure 1: System overview use case



### List of use case

#### <Guest> Overview Use Case

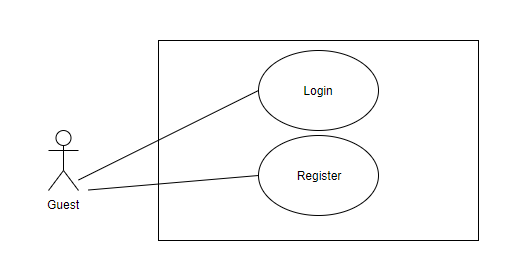


Figure 2: <Guest> Overview use case

##### <Guest> Login



Figure 3 <Guest> Login

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_01** | | | |
| **Use Case No.** | **EMP\_UC\_01** | Use Case Version | 1.1 |
| **Use Case Name** | Login | | |
| **Author** | Nguyễn Hữu Thắng | | |
| **Date** | 15/07/2019 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows user to login to the system.   **Goal:**   * User can login to the system   **Trigger:**   * User send the login command. * User clicks on “Sign-in” button   **Precondition:**   * User is viewing the login screen   **Post Conditions:**   * Success: User login to the system successfully * Fail: Show error messages   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User go to the login view | The system requires the following input from User includes:   * Username: text, required * Password: text, required | | 2 | User inputs information. |  | | 3 | User sends command to login | The system show message that the user has logged in successfully  [Exception 1] |   **Exception 1:**   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationship:** N/A.  **Business rule:**   * User password character is secured and hidden with special character * The system generates an access token * The system generates a refresh token * System can use access token to verify identity without reenter username and password again * Refresh token can be used to retrieve new access token with no username or password needed * Access token have expiration time of 15 mins * Refresh token has expiration time of 90 days * Refresh token and access token is signed with RSA256 algorithm * Refresh token and access token also contain at least following information:   + Username   + User id   + Expire time   + Created time   + Signature for security * User is required a new refresh token if password changed or refresh token is expired | | | |

Table :<Guest> Login

##### <Guest> Register



Figure 4: <Guest> Register

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE - EMP\_UC\_02 | | | |
| Use Case No. | EMP\_UC\_02 | Use Case Version | 1.1 |
| Use Case Name | Register | | |
| Author | Nguyễn Hữu Thắng | | |
| Date | 15/07/2019 | Priority | Normal |
| Actor:   * Guest   Summary:   * This use case allows user to register to the system.   Goal:   * User can register to the system   Trigger:   * User send the register command. * User clicks “Sign up”   Precondition:   * User is viewing the register screen   Post Conditions:   * Success: User register to the system successfully * Fail: Show error messages   Main Success Scenario:   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User go to the guess view | The system requires the following input from User includes:   * Username: text, required * Password: text, required * Confirm password: text, required * Name: text, required * Email: text, required * Phone: text as number, required * Profile image: image, required | | 2 | User inputs information. |  | | 3 | User sends the command to register | The system show message that the user has registered successfully  [Exception 1] |   Exception 1:   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   Relationship: N/A.  Business rule:   * User password must not be saved directly plain text, and must be hashed * The system validates user input with the following constraints:   + Username: unique   + Phone number: number format   + Email: as an email format   + The confirm password matches the password | | | |

Table :<Guest> Register

#### D:\FPT\Capstone2019\EmailMarketing\Documents\Diagram\MindSendingUseCase.png<Registered User> Overview Use Case

Figure 5: <Registered User>Overview use case

##### <Registered User> Logout



Figure 6: <Registered User> Logout

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - EMP\_UC\_03** | | | |
| **Use Case No.** | **EMP\_UC\_03** | **Use Case Version** | 1.1 |
| **Use Case Name** | Logout | | |
| **Author** | Nguyễn Hữu Thắng | | |
| **Date** | 15/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to logout to the system.   **Goal:**   * User can logout from the system   **Trigger:**   * User send a logout command.   **Precondition:**   * User is viewing the account screen   **Post Conditions:**   * Success: User logout of the system successfully * Fail: Show error messages   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User go to the account view |  | | 2 | User send a logout command. | The system show message that the user has logged out successfully  [Exception 1] |   **Exception 1:**   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationship:** N/A.  **Business rule:**   * After logged out, user access the system as role “Guest” | | | |

Table :<Registered User> Log out

##### <Registered User> View profile

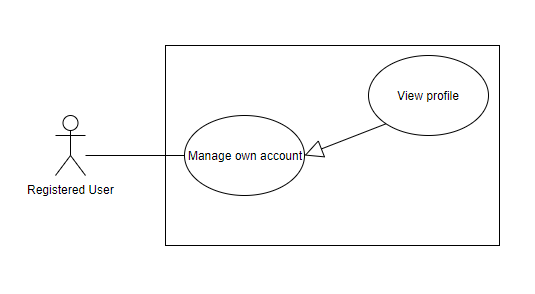


Figure 7: <Registered User> View User Profile

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - EMP\_UC\_04** | | | |
| **Use Case No.** | **EMP\_UC\_04** | **Use Case Version** | 1.0 |
| **Use Case Name** | Manage User Profile | | |
| **Author** |  | | |
| **Date** | 15/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to manage own account   **Goal:**   * User can edit their own account   **Triggers:**   * User clicks “User profile”. * User send manage account command to the system.   **Preconditions:**   * User must login into the system with role Registered User   **Post conditions:**   * Success: The profile successful updated. * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to manage account view | The system requires input from user includes:   * Username: text, * First Name: text, * Last Name:text, * Email Address: text, * City: text, * Profile image: image, * Country: text, * Information: text, * Password: text, password | | 2 | User inputs information. |  | | 3 | User send edit profile command. | System shows message profile has been edited successfully.  [Exception 1]  [Exception 2] |   **Exception 1:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Edit profile fail. (Input contain special characters) | The system shows an error message “Input may not contain special characters” |   **Exception 2:**   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | 1 | When the connection to the server is lost | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A.  **Business Rules:**   * After editing profile, Registered User must be verified again by staff * Registered User’s status change into “Not verified” | | | |

Table :<Registered User> View profile

##### <Registered User> Edit Profile

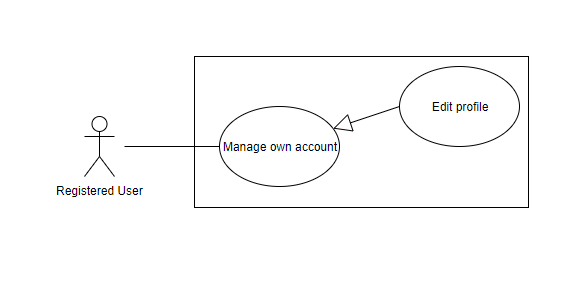


Figure 8: <Registered User> Edit Profile

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_5** | | | |
| **Use Case No.** | **EMP\_UC\_5** | **Use Case Version** | 1.1 |
| **Use Case Name** | Edit profile | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to edit profile   **Goal:**   * User edit profile successfully   **Triggers:**   * User send command to edit profile   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: profile edit successfully * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User view profile before edit | System show profile on screen | | 2 | User edit profile |  | | 3 | User update edited profile | System show message “profile edited successfully” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view the profile before edit * User can update the profile later | | | |

Table :<Registered User> Edit profile

##### <Registered User> View Report



Figure 9: <Register User> View Report

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - EMP\_UC\_6** | | | |
| **Use Case No.** | **EMP\_UC\_6** | **Use Case Version** | 1.0 |
| **Use Case Name** | View report | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to view report   **Goal:**   * User can view the report by week, month   **Triggers:**   * User send view report command. * User clicks on ”Dashboard”   **Preconditions:**   * User must login into the system with role Registered User.   **Post conditions:**   * Success: The report display on screen * Fail: The report cannot displayed on screen   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to dashboard | System display:  + The report by the latest campaign  + List contact by type  + List of group have been recently created  + List of contact have been recently added |   **Exception:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | When the connection to the server is lost | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A.  **Business Rules:**   * User can view the report of sent email, email pending by week, month * User can analyses the report | | | |

Table :<Registered User> View Report

##### <Registered User> View Contact

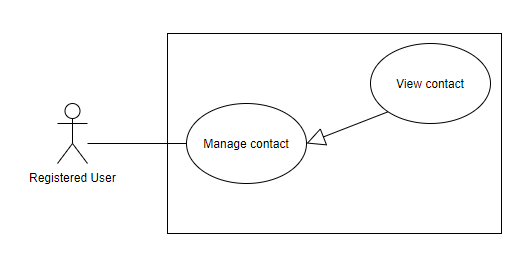


Figure 10: <Registered User> View Contact

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - EMP\_UC\_7** | | | |
| **Use Case No.** | **EMP\_UC\_7** | **Use Case Version** | 1.0 |
| **Use Case Name** | View contact | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows registered user to view all contacts   **Goal:**   * User can view all the list of contact   **Triggers:**   * User click “Contact” to see the contact list   **Preconditions:**   * User must login into the system with role Registered User.   **Post conditions:**   * Success: List all contact. * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User sends command to view the contact list | * The system show list of all contact [Exception 1] |   **Exception 1:**   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | 1 | When the connection to the server is lost | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A.  **Business Rules:**   * User can view all the information related to him, created in the past or in the present * Each contact has one of the following statuses at a time:   + New subscriber: contact is newly created   + Subscriber: contact is added for a time * User can see the following information about each contact:   + Email   + First name   + Last name   + Phone number   + Address   + Date of birth | | | |

Table <Registered User> View contact

##### <Registered User> Add Contacts

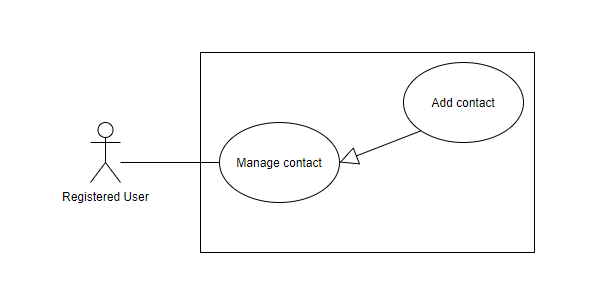


Figure 11: <Registered User> Add Contacts

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE - EMP\_UC\_8** | | | |
| **Use Case No.** | **EMP\_UC\_8** | **Use Case Version** | 2.0 |
| **Use Case Name** | Add contacts | | |
| **Author** |  | | |
| **Date** | 14/06/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows Registered User to add a contact   **Goal:**   * User successfully add contacts   **Triggers:**   * User send add contact command to the system. * User clicks on “Add contact”   **Preconditions:**   * User must login into the system with role Registered User.   **Post conditions:**   * Success: New contact will be add to the default list * Fail: Show error messages   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to add contact screen | Display a table show all contacts with information:   * Email Address : @email.com * Status: * First Name * Last Name * Action | | 2 | User choose add contacts button | Display two option:   * Upload CSV file * Manual Add | | 3 | User chooses to upload CSV file | * Display two options: * Add contacts * Add contacts and include in an existing list   Choose CSV file to upload | | 4 | User chooses manual add | * Display three option: * Add contacts * Add contacts and include in an existing list | | 4.1 | User fill in contact info | * Display six field to fill * First name * Last name * Email * Phone number * Address * Date of birth | | 4.2 | User add more contact | * System add more six field   [Exception 1] | | 4.3 | User save contact | * System save new contact   [Exception 1]  [Exception 2] |   **Exception 1:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | When the contact is already in the list | System shows message the "This email is existed" when the contact was added to the list. |   **Exception 2:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 2 | When the connection to the server is lost | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A.  **Business Rules:**   * Contact only contain related information * Contact can later update their information. * Contact is required for Email, First name, Last name. * A contact can have multiple group. * A contact contains the following information:   + Email   + First name   + Last name   + Phone number   + Address   + Date of birth | | | |

Table :<Registered User> Add Contacts

##### <Registered User> Delete Contacts

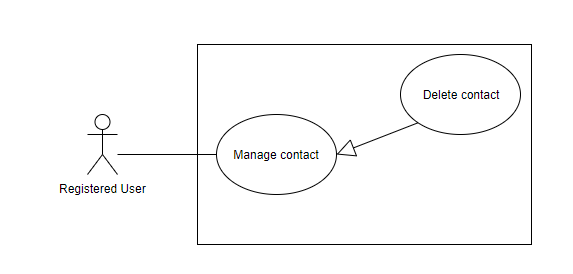


Figure 12: <Registered User> Delete contacts

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_9** | | | |
| **Use Case No.** | **EMP\_UC\_9** | **Use Case Version** | 2.0 |
| **Use Case Name** | Delete contacts | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Authenticated Registered User   **Summary:**   * This use case allows User to delete contacts.   **Goal:**   * User successfully delete contacts in the system.   **Triggers:**   * User send delete command to the system.   **Preconditions:**   * User must login into the system with role Registered User. * Contact does not belong to any campaigns   **Post conditions:**   * Success: contacts is removed by user. * Fail: Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to manage contacts screen. | Display a table show list of contacts. | | 2 | User send delete contacts command. | System start removing contacts.  [Exception 1]  [Exception 2]  System show message remove successful |   **Exception 1:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Removed fail. | Show message to notify User that remove failed. |   **Exception 2:**   |  |  |  | | --- | --- | --- | | **Step** | **Cause** | **System Response** | | 1 | When the connection to the server is lost | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * In case of success scenarios, contacts will be removed from the system. | | | |

Table <Registered User> Delete contacts

##### <Registered User> Search Contact

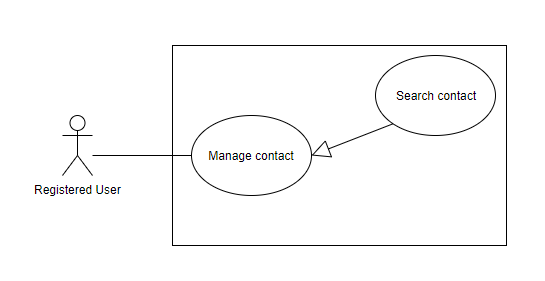


Figure 13: <Registered User> Search Contact

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_10** | | | |
| **Use Case No.** | **EMP\_UC\_10** | **Use Case Version** | 1.1 |
| **Use Case Name** | Search Contact | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to search contact by typing keyword   **Goal:**   * User search contact   **Triggers:**   * User send command to search contact   **Preconditions:**   * User must login to the system * The keyword must be included   **Post conditions:**   * Success: User can search contact * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send command to search contact | System ask user to input keyword to search | | 2 | User input the keyword | System show the lists that match keyword |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User input wrong keyword | System do not show anything |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User will get the search results list. | | | |

Table <Registered User> Search contact

##### <Registered User> View Templates

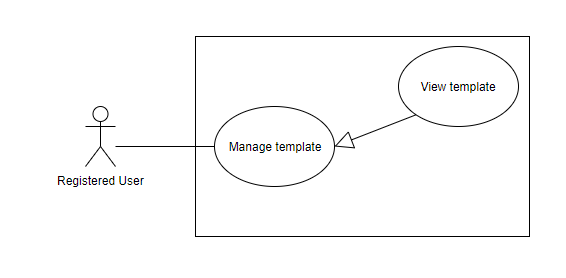


Figure 14: <Registered User> View Templates

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_11** | | | |
| **Use Case No.** | **EMP\_UC\_11** | **Use Case Version** | 2.0 |
| **Use Case Name** | View templates | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows registered user to view all templates   **Goal:**   * User can view all of templates   **Triggers:**   * User click to templates to see templates   **Preconditions:**   * User must login into the system with role Registered User.   **Post conditions:**   * Success: List all templates. * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to Template | System show a list of Custom templates and Mindsending templates  [Exception 1] |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view all the templates in the system: * Custom templates * Mindsending templates * User can preview template before choose * Action button to duplicate or edit template is allowed authority:   + User can only edit their own template | | | |

Table <Registered User> View templates

##### <Registered User> Create Template

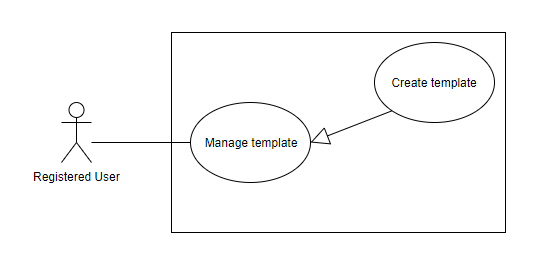


Figure 15: <Registered User> Create template

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_12** | | | |
| **Use Case No.** | **EMP\_UC\_12** | **Use Case Version** | 2.0 |
| **Use Case Name** | Create template | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered user   **Summary:**   * This use case allows user to create new template   **Goal:**   * User can create new template   **Triggers:**   * User send command to create new template   **Preconditions:**   * User must login to the system with role Registered User   **Post conditions:**   * Success:   + New template will be added to the system * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to create new template | System suggest two option:   * Export HTML * Load HTML | | 2 | User save template | System show message “Template is created successfully”  [Exception ] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can review the template before save. * User can export HTML or load HTML on device to system | | | |

Table <Registered User> Create templates

##### <Registered User> Edit Template

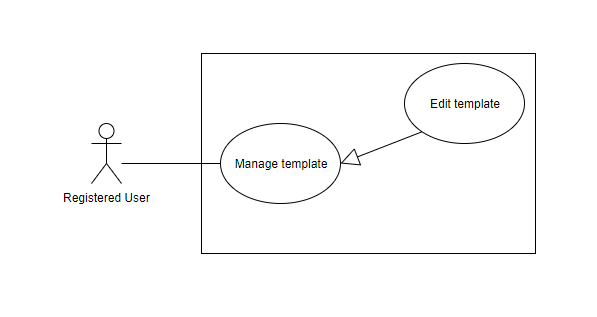


Figure 16: <Registered User> Edit template

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_13** | | | |
| **Use Case No.** | **EMP\_UC\_13** | **Use Case Version** | 1.1 |
| **Use Case Name** | Edit template | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to edit template   **Goal:**   * User edit template successfully   **Triggers:**   * User send command to edit template   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: Template edit successfully * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User view template before edit | System show template on screen | | 2 | User edit template |  | | 3 | User save edited template | System show message “ Template edited successfully” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view the template before edit | | | |

Table <Registered User> Edit template

##### 2.3.2.12 <Registered User> Duplicate Template

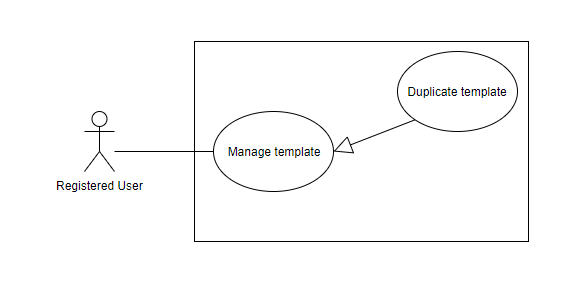


Figure 17: <Registered User> Duplicate Template

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_14** | | | |
| **Use Case No.** | **EMP\_UC\_14** | **Use Case Version** | 1.0 |
| **Use Case Name** | Duplicate template | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to duplicate template   **Goal:**   * User can duplicate template   **Triggers:**   * User click into duplicate button   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: Template is duplicated * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User sends command to duplicate template | System show duplicate pop-up | | 2 | User fill template name to duplicate |  | | 3 | User click save | System show message “Duplicate template successfully” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * By this action, user can view all contact lists. * The default behavior of duplication is to execute the action and automatically copy values from the current template into the new template with the new name | | | |

Table <Registered User> Duplicate Template

##### 2.3.2.13 <Registered User> View Group

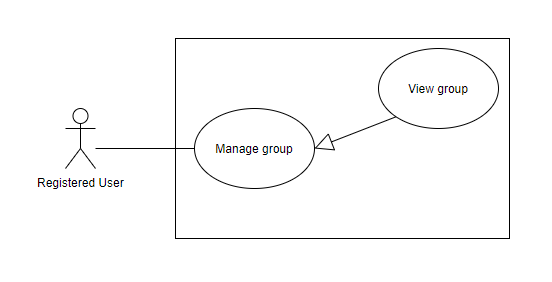


Figure 18: <Registered User> View Group

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_15** | | | |
| **Use Case No.** | **EMP\_UC\_15** | **Use Case Version** | 1.0 |
| **Use Case Name** | View Group | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to manage Group   **Goal:**   * User can manage Group   **Triggers:**   * User click into Group   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: Contact Group display on screen * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to Group screen | System show a list of campaign  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view all the group related to him/her, created in the past or in the present * Each group included to one or many contact * User can see the following information about each campaign:   + Group name  + Description  + Contacts  + Action   * Action button to edit or delete group is allowed authority * User can choose to view group detail related to each group | | | |

Table <Registered User> View group

##### 2.3.2.14 <Registered User> Edit Group

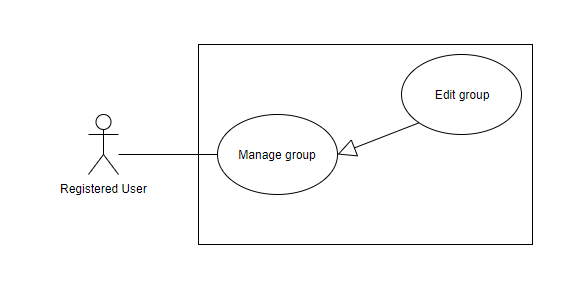


Figure 19: <Registered User> Edit group

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_16** | | | |
| **Use Case No.** | **EMP\_UC\_16** | **Use Case Version** | 1.1 |
| **Use Case Name** | Edit group | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to edit group   **Goal:**   * User edit group successfully   **Triggers:**   * User send command to edit group   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: group edit successfully * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User view group before edit | System show group on screen | | 2 | User edit group |  | | 3 | User save edited group | System show message “ group edited successfully” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view the group before edit | | | |

Table <Registered User> Edit group

##### 2.3.2.15 <Registered User> Create Group

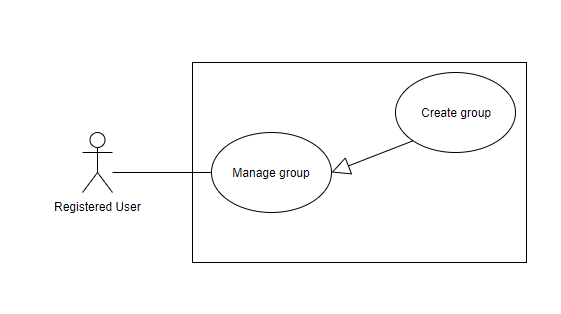


Figure 20: <Registered User> Create Group

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_17** | | | |
| **Use Case No.** | **EMP\_UC\_17** | **Use Case Version** | 1.0 |
| **Use Case Name** | Create Group | | |
| **Author** |  | | |
| **Date** | 16/07/2018 | **Priority** | Normal |
| **Actor:**   * Registered user   **Summary:**   * This use case allows user to create Group   **Goal:**   * User create lists successfully   **Triggers:**   * User send command to create Group   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success:   + This Group is added successfully * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to create Group | System show create Group pop-up | | 2 | User fill Group name and description to create |  | | 3 | User click create | System show message “Add group successfully” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Group only contain related information * Group can later update their information. * Contact is required for Name and description. * A group can have multiple contact. * A group contains the following information:   + Name   + Description   + Contacts   + Action | | | |

Table <Registered User> Create group

##### 2.3.2.16 <Registered User> Search Group

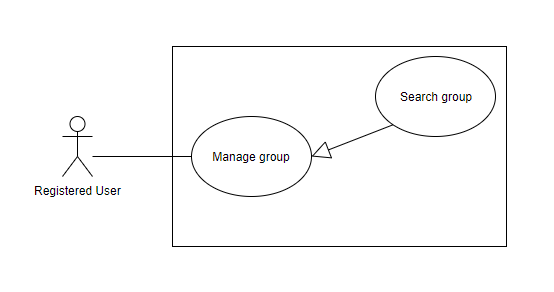


Figure 21: <Registered User> Search Group

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_18** | | | |
| **Use Case No.** | **EMP\_UC\_18** | **Use Case Version** | 1.1 |
| **Use Case Name** | Search Group | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to search lists by typing keyword   **Goal:**   * User search Group   **Triggers:**   * User send command to search Group   **Preconditions:**   * User must login to the system * The keyword must be included   **Post conditions:**   * Success: User can search Group * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send command to search Group | System ask user to input keyword to search | | 2 | User input the keyword | System show the lists that match keyword |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User input wrong keyword | System donot show anything |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User will get the search results list. | | | |

Table <Registered User> Search Group

##### 2.3.2.17 <Registered User> Delete Group

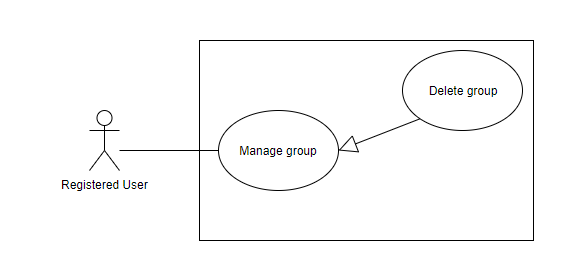


Figure 22: <Registered User> Delete Group

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_19** | | | |
| **Use Case No.** | **EMP\_UC\_19** | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Group | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered user   **Summary:**   * This use case allows user to delete Group   **Goal:**   * User delete Group successfully   **Triggers:**   * User send command to delete Group   **Preconditions:**   * User must login to the system * Group does not belong to any campaigns   **Post conditions:**   * Success:   + The request status changed to “Accepted” * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to delete Group | System ask user to confirm the action | | 2 | User choose to confirm the accept action [Alternative 1] | System show message that the request is approved  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * In case of success scenarios, group will be removed from the system. | | | |

Table <Registered User> Delete Group

##### 2.3.2.18 <Registered User> View Appointment

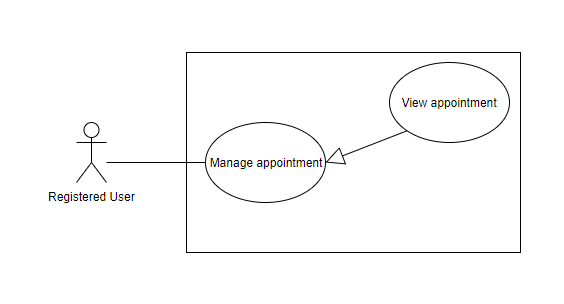


Figure 23: <Registered User> View Appointment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_20** | | | |
| **Use Case No.** | **EMP\_UC\_20** | **Use Case Version** | 1.0 |
| **Use Case Name** | View Appointment | | |
| **Author** |  | | |
| **Date** | 17/06/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to view appointment   **Goal:**   * User can view all appointment   **Triggers:**   * Supplier send command to show all appointment   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: System will show all appointment on screen * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to show all appointment | System show a list of appointment  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view all the appointment related to him/her, created in the past or in the present * Each appointment include one or many group * User can see the following information about each appointment:   + Name  + Start on  + Invitations sent  + Invitations sent  + Registrants  + Action   * User can choose to view appointment detail related to each appointment | | | |

Table <Registered User> View appointment

##### 2.3.2.19 <Registered User> Create Appointment

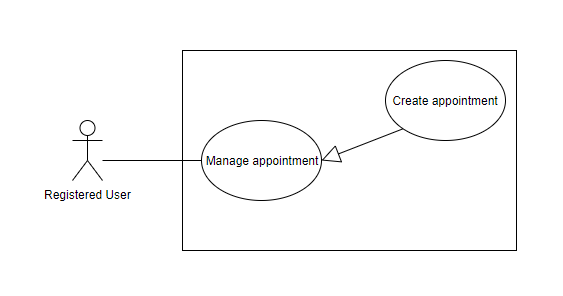


Figure 24: <Registered User> Create appointment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_21** | | | |
| **Use Case No.** | **EMP\_UC\_21** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create appointment | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to create appointment   **Goal:**   * User can create appointment   **Triggers:**   * User send command to create appointment   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: New appointment is added * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to appointment | System show all appointment with   * Name * Start on * Invitations sent * Registrants * Action | | 2 | User click to create appointment | System show message to confirm create appointment | | 3 | User input appointment information | System show appointment information to input:   * To * From * Subject * Content | | 4 | User confirm the action  [Alternative 1] | System display successful message  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User input |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. | | 2 | User input invalidate information. | System show error message for the input mistake |   **Relationships:** N/A  **Business Rules:**   * Appointment only contain related information * Appointment is required for Group contact, Sender name, Email Address, subject and content. * An Appointment can have multiple group. * An Appointment contains the following information: * Name * Start on * Invitations sent * Registrants * Action | | | |

Table <Registered User> Create Appointment

##### 2.3.2.20 <Registered User> View Embed Form

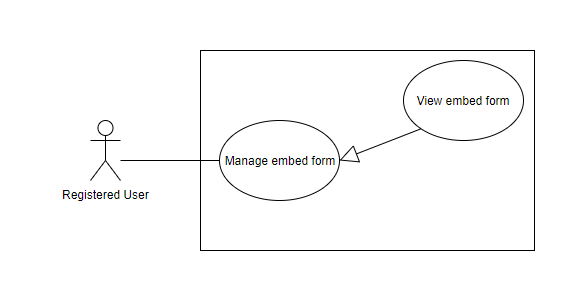


Figure 25: <Registered User> View Embed Form

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_22** | | | |
| **Use Case No.** | **EMP\_UC\_22** | **Use Case Version** | 1.0 |
| **Use Case Name** | View Embed Form | | |
| **Author** |  | | |
| **Date** | 17/06/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to view Embed Form   **Goal:**   * User can view all Embed Form   **Triggers:**   * Supplier send command to show all Embed Form   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: System will show all Embed Form on screen * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to show all campaigns | System show a list of Embed Form  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view all the Embed Form related to him/her, created in the past or in the present * Each Embed Form include one group * User can see the following information about each Embed Form:   + Form name  + Create on  + Group contact  + Action   * User can choose to view Embed Form detail related to each Embed Form * User can choose to copy code from Embed Form that | | | |

Table <Registered User> View Embed form

##### 2.3.2.21 <Registered User> Create Embed Form

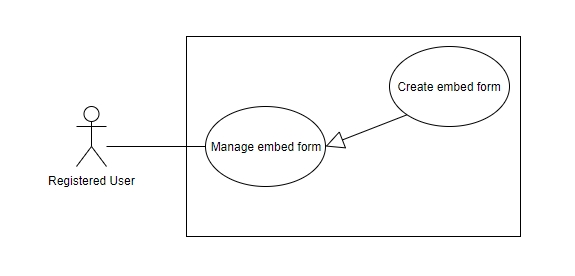


Figure 26: <Registered User> Create Embed Form

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_23** | | | |
| **Use Case No.** | **EMP\_UC\_23** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create embed form | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to create embed form   **Goal:**   * User can create embed form   **Triggers:**   * User send command to create embed form   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: New embed form is added * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to embed form | System show all embed form with   * Form name * Create on * Group contact * Action | | 2 | User click to create embed form | System show message to confirm create embed form | | 3 | User setting embed form information | System show embed form setting to adjust:   * Form name * Group * Add custom field | | 4 | User confirm the action  [Alternative 1] | System display successful message  [Exception 1] |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. | | 2 | User input invalidate information. | System show error message for the input mistake |   **Relationships:** N/A  **Business Rules:**   * Embed form only contain related information * Embed form can later update their information. * Embed form is required for Form name, group contact, and custom field. * An Embed form can only has 1 group. * An Embed form contains the following information:   + Form name   + Create on   + Group contact   + Action | | | |

Table <Registered User> Create Embed form

##### 2.3.2.22 <Registered User> View regular Campaigns

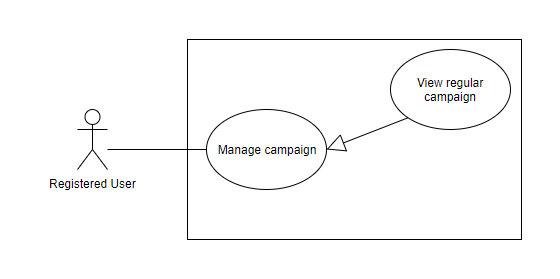


Figure 27: <Registered User> View Campaigns

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_24** | | | |
| **Use Case No.** | **EMP\_UC\_24** | **Use Case Version** | 1.0 |
| **Use Case Name** | View Campaigns | | |
| **Author** |  | | |
| **Date** | 17/06/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to view campaigns   **Goal:**   * User can view all campaigns   **Triggers:**   * Supplier send command to show all campaigns   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: System will show all campaigns on screen * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to show all campaigns | System show a list of campaign  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view all the regular campaign related to him/her, created in the past or in the present * Each campaign include one or many group * Each campaign has one of the following statuses at a time:   + Draft: mail is not sent  + Sending: mail is sending  + Done: mail is sent   * User can see the following information about each campaign:   + Status  + Campaign name  + Delivery  + Opens  + Clicks  + Action   * Action button to change campaign status is allowed authority * User can choose to view campaign detail related to each campaign * User can choose to duplicate campaign | | | |

Table <Registered User> View Campaigns

##### 2.3.2.23 <Registered User> Create Regular Campaigns

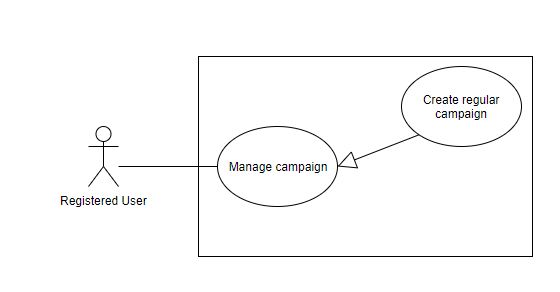


Figure 28: <Registered User> Create regular campaigns

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_25** | | | |
| **Use Case No.** | **EMP\_UC\_25** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create regular campaigns | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to create regular campaigns   **Goal:**   * User can create regular campaigns   **Triggers:**   * User send command to create regular campaigns   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: New campaigns is added * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User goes to regular campaigns | System show all campaigns and campaign statistics   * Status * Campaign Name * Clicks * Opens * Unsubscribe | | 2 | User click to create campaign | System show message to confirm create campaigns | | 3 | User input campaigns information | System show campaigns information to input:   * To * From * Subject * Content | | 4 | User confirm the action  [Alternative 1] | System display successful message  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User input |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. | | 2 | User input invalidate information. | System show error message for the input mistake |   **Relationships:** N/A  **Business Rules:**   * Campaign only contain related information * Campaign can later update their information. * Campaign is required for Group contact, Sender name, Email Address, subject and content. * A campaign can have multiple group. * A campaign contains the following information:   + status   + campaign name   + delivery   + open   + click   + Action | | | |

Table <Registered User> Create Regular Campaigns

##### 2.3.2.24 <Registered User> Edit Regular Campaigns

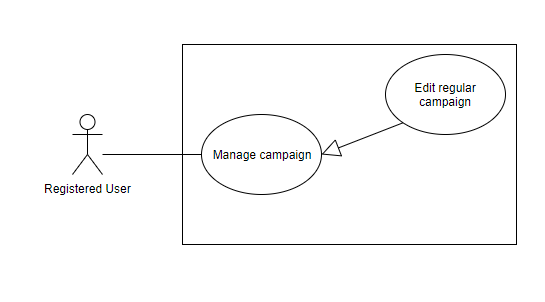


Figure 29: <Registered User> Edit Regular Campaigns

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_26** | | | |
| **Use Case No.** | **EMP\_UC\_26** | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Campaigns | | |
| **Author** |  | | |
| **Date** | 18/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to edit campaigns   **Goal:**   * User can edit campaigns   **Triggers:**   * User send command to edit campaigns   **Preconditions:**   * User must login to the system * Campaign status is Draft   **Post conditions:**   * Success: The campaigns is edited * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User go to editing campaigns | System requires information about campaigns to edit, which contains:   * Group * To * From * Subject * Content | | 2 | User input information |  | | 3 | User send command to save new campaign | System validate information, display request for confirmation | | 4 | User confirm the action  [Alternative 1] | System display successful message  [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User choose not to confirm the action | System hides the message and go back to inputting view |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. | | 2 | User input invalidate information. | System show error message for the input mistake |   **Relationships:** N/A  **Business Rules:**   * User can view the campaign before edit * User cannot edit campaign with “Done” status * Further information can be found in Use Case <EMP\_UC\_19> (Create regular campaign) | | | |

Table <Registered User> Edit Campaigns

##### 2.3.2.25 <Registered User> View Workflow

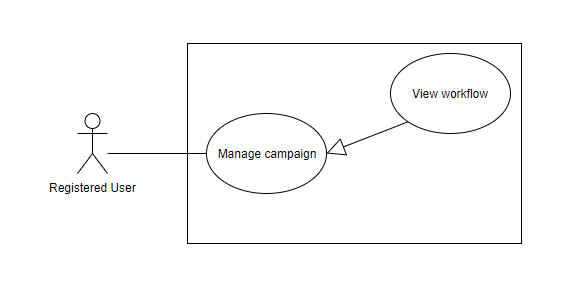


Figure 30: <Registered User> View Workflow

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_27** | | | |
| **Use Case No.** | **EMP\_UC\_27** | **Use Case Version** | 1.0 |
| **Use Case Name** | View Workflow | | |
| **Author** |  | | |
| **Date** | 17/06/2019 | **Priority** | High |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to view workflow   **Goal:**   * User can view all workflow   **Triggers:**   * Supplier send command to show all workflow   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: System will show all workflow on screen * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to show all workflow | System show a list of workflow  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * User can view all the workflow related to him/her, created in the past or in the present * Each workflow include one form and many tasks * User can see the following information about each workflow:   + Status  + Campaign name  + Start on  + Action   * Action button to change campaign status is allowed authority * User can choose to view workflow detail related to each workflow * User can choose to pause workflow | | | |

Table <Registered User> View workflow

##### 2.3.2.26 <Registered User> Create Workflow

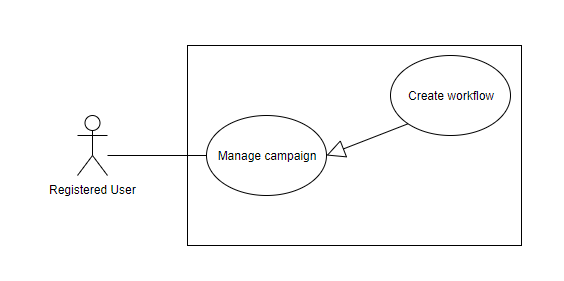


Figure 31: <Registered User> Create Workflow

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_28** | | | |
| **Use Case No.** | **EMP\_UC\_28** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create Workflow | | |
| **Author** |  | | |
| **Date** | 16/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to create workflow   **Goal:**   * User can create Workflow   **Triggers:**   * User send command to create workflow   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: The workflow is created * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to create workflow | System show pop-up to fill in workflow name | | 2 | User choose the form | System show all form for user to choose | | 3 | User choose the gateway | System show conditions for gateway are:  + Clicked  + Opened | | 4 | User choose task | System show all campaigns and appointments for user to choose | | 5 | User choose to confirm the approve action  [Alternative] | System show message that the workflow is created |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User choose not to confirm the action | System hides the message and does nothing |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Workflow only contain related information * Workflow is required for Form contact, Gateway and Tasks. * A workflow can have multiple task. * A workflow can only have a form. | | | |

Table <Registered User> Create workflow

##### <Registered User> Create Segment

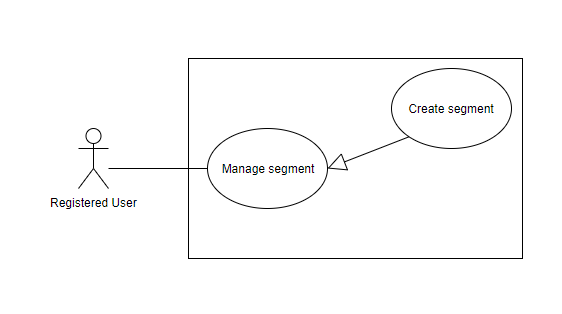


Figure 32: <Registered User> Create Segment

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_29** | | | |
| **Use Case No.** | **EMP\_UC\_29** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create Segment | | |
| **Author** |  | | |
| **Date** | 27/07/2019 | **Priority** | Normal |
| **Actor:**   * Registered User   **Summary:**   * This use case allows user to create segment   **Goal:**   * User can create segment   **Triggers:**   * User send command to create segment   **Preconditions:**   * User must login to the system   **Post conditions:**   * Success: The segment is created * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to create segment | System display two options for finding condition:  + Any  + All | | 2 | User choose the contact details | System show:   * Name * Email * Birthday * Address * Subscription date * Engagement score * Group | | 3 | User choose the contact actions | System show :   * Mail not opened * Mail opened * Mail not clicked * Mai clicked | | 4 | User press apply button | System show all contact that match the segmentation | | 5 | User choose to confirm the approve action  [Alternative] | System show message that the segment is created |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | User choose not to confirm the action | System hides the message and does nothing |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Segment only display contacts that match the condition * Segment will be saved as a new contact group * A segment can have multiple conditions.. | | | |

Table <Registered User> Create segment

#### <Admin guest> Overview Use Case

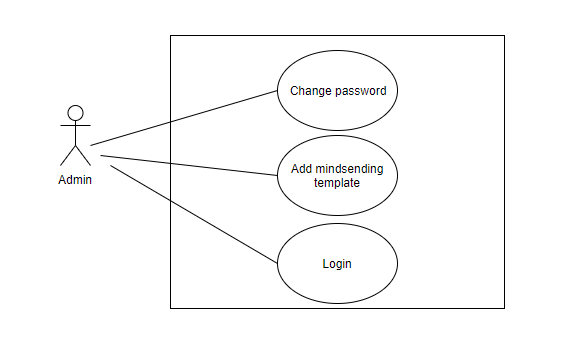


Figure 33: <Admin>Overview use case

##### <Admin> Login to web admin



Figure 34: <Admin>Login to web admin

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_30** | | | |
| **Use Case No.** | **EMP\_UC\_30** | **Use Case Version** | 1.1 |
| **Use Case Name** | Login to web admin | | |
| **Author** |  | | |
| **Date** | 20/07/2019 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows guest to log in the system.   **Goal:**   * Guest can log in the system with specific role.   **Triggers:**   * Guest sends login command.   **Preconditions:** N/A  **Post conditions:**   * Success: Guest is exactly authorized with specific role in the system. * Fail: System shows error messages.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest go to login screen. | Login screen is shown with following labels and fields:   * Username: text input, required. * Password: text input, required. | | 2 | Guest inputs information. |  | | 3 | Guest sends command to login to system. | Guest logins system with his specific role. [Alternative 1] [Exception 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Guest enter wrong identity information. | System shows error message “Wrong username or password. Please try again.” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Login failed. | Show message to notify admin that login failed. |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted before being sent to server. | | | |

Table <Admin> Login to web admin

##### 2.3.3.2 <Admin> Change user password



Figure 35: <Admin>Change user password

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_31** | | | |
| **Use Case No.** | **EMP\_UC\_31** | **Use Case Version** | 1.1 |
| **Use Case Name** | Change user password | | |
| **Author** |  | | |
| **Date** | 20/07/2019 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to reset user password by email.   **Goal:**   * Guest can reset password by email.   **Triggers:**   * Guest sends reset password command.   **Preconditions:** N/A  **Post conditions:**   * Success: User’s password successful update in database and send to user’s email. * Fail: User password cannot be updated.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | Guest go to forgot your password screen. | Forgot your password screen is shown with following labels and fields:   * Email: text input, required. | | 2 | Guest inputs information. |  | | 3 | Guest sends command to reset password to system. | Update user password in the system.  [Alternative 1]  [Exception 1]  System shows message reset password has been updated successfully |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **No** | **Actor Action** | **System Response** | | 1 | Guest enter wrong identity information. | System shows error message” Wrong email.” |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | Reset password failed. | Show message to notify admin that reset password failed. |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted before being sent to server. * Email have been registered before. | | | |

Table <Admin> Login to web admin

##### <Admin> Add MindSending template

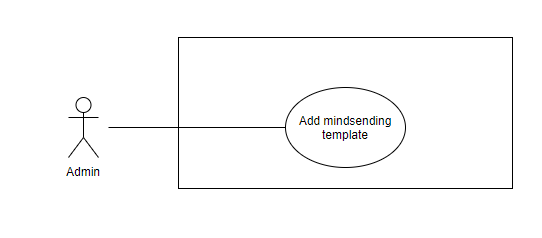


Figure 36: <Admin> Add MindSending template

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – EMP\_UC\_32** | | | |
| **Use Case No.** | **EMP\_UC\_32** | **Use Case Version** | 2.0 |
| **Use Case Name** | Add mindsending template | | |
| **Author** |  | | |
| **Date** | 26/07/2019 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin to create new mindsending template   **Goal:**   * Admin can create new mindsending template   **Triggers:**   * Admin send command to create new mindsending template   **Preconditions:**   * Admin must login to the system with role Admin   **Post conditions:**   * Success:   + New template will be added to the system * Fail: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor Action** | **System Response** | | 1 | User send the command to create new template | System suggest two option:   * Export HTML * Load HTML | | 2 | User save template | System show message “Add template successfully”  [Exception ] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **No** | **Cause** | **System Response** | | 1 | No internet connection | System shows message the "Please check your connection" when the internet is lost. |   **Relationships:** N/A  **Business Rules:**   * Admin can review the template before save. * Admin can export HTML or load HTML on device to system | | | |

Table <Registered User> Add mindsending templates

## Software System Attribute

### Usability

Website’s UI is fit for each browser on each device

* Font style: Cambria, Sans-serif
* Font size: 12px -26px
* Color: green, black, red, white, blue, Light Slate Grey, Fuego, Honeysuckle…
* Background: White, Catalina Blue, Whisper…

### Reliability

* The information storing on the database is permanent.

### Availability

* The system must be available at any time 24/7.

### Security

* Only Manager has fully access to the system
* Input data is validated before saving to database.

### Maintainability

* The system is divided into separated modules for easy maintain.

### Portability

* Web application can be run on Chrome browser version 42 or later

### Performance

* System has successfully test with basic standard

## Conceptual Diagram

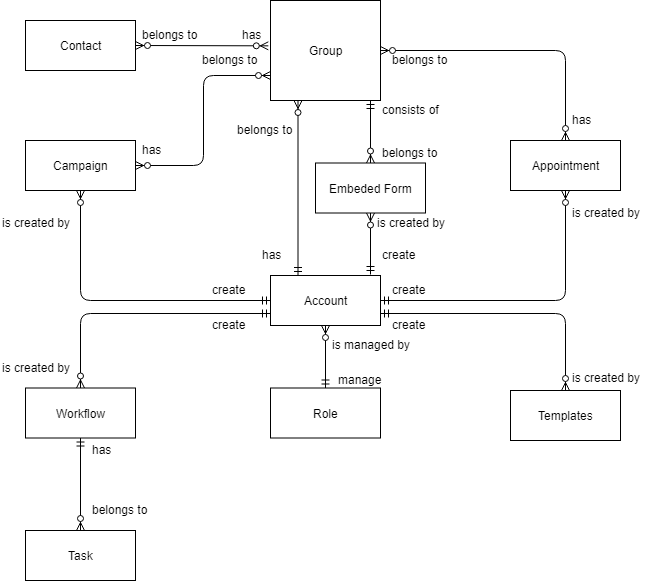


Figure 4 Conceptual Diagram

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe content of all entities** | |
| **Entity name** | **Description** |
| Account | Contains the user’s information. |
| Contact | Contains the contact’s information. |
| Campaign | Contains the campaign’s information. |
| Template | Contains the template’s information. |
| Appointment | Contains the appointment’s information. |
| Embed form | Contains the embed form’s information. |
| Group | Contains the group’s information. |
| Workflow | Contains the workflow’s information. |
| Task | Contains the task’s information. |
| Role | Contains the role’s information. |

Table 40 Conceptual Diagram Dictionary

# Software Design Description

## Design Overview

This document describes the technical and user interface design of EMP system. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

The detailed design describes static and dynamic structure for each component and functions. It includes class diagram EMP, class explanations and sequence diagram EMP for each use cases.

 The database design describes the relationships between entities and details of each entity.

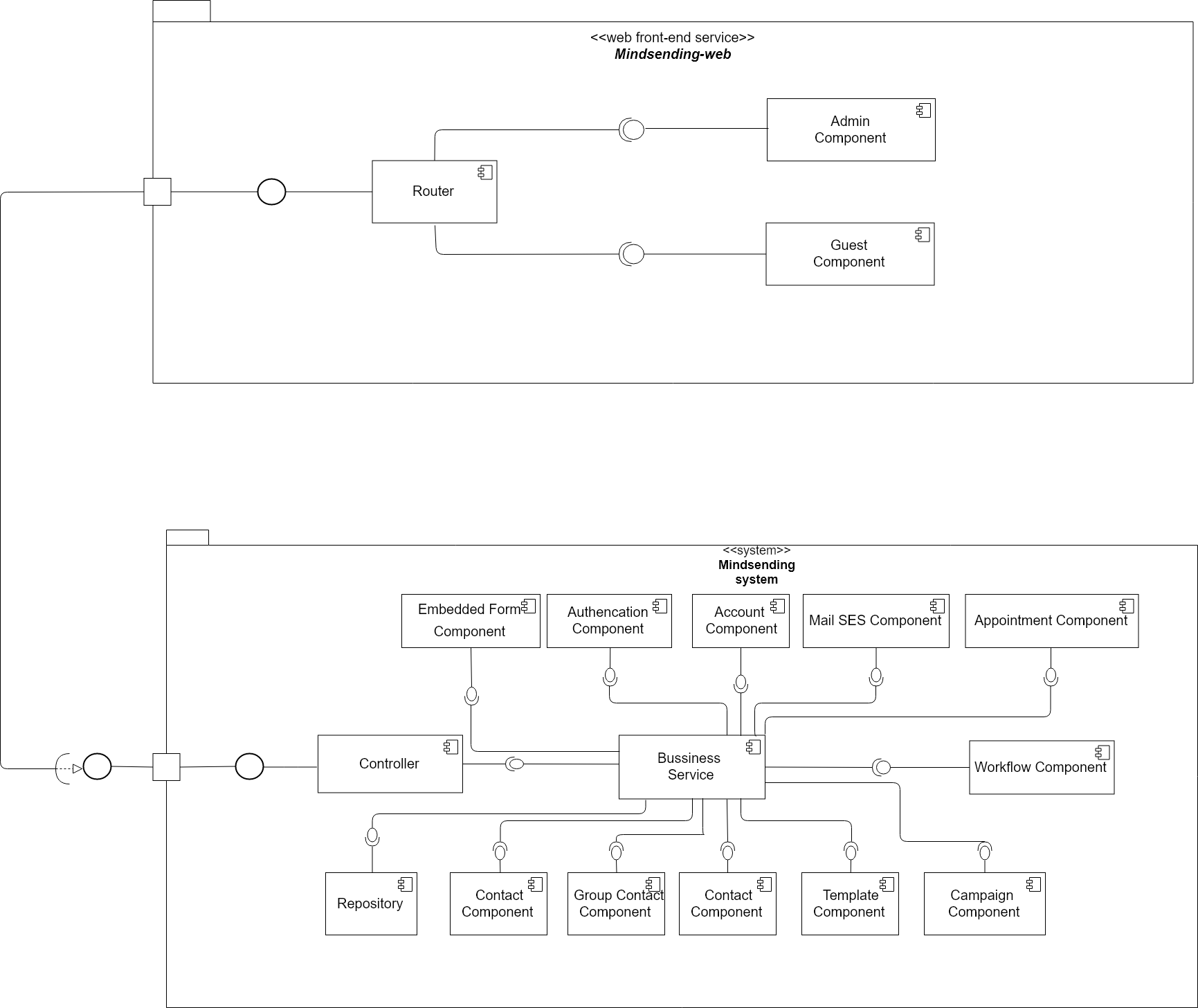
Document overview:

* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagram EMP that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe screens design.
* Section 6: describe a fully attributed ERD.
* Section 7: describe algorithm EMP.

## C:\Users\ThangNguyen\Downloads\68412524_589431241461426_7489382843403993088_n.jpgSystem Architecture Design

Figure 5 System Architecture diagram

## Component Diagram



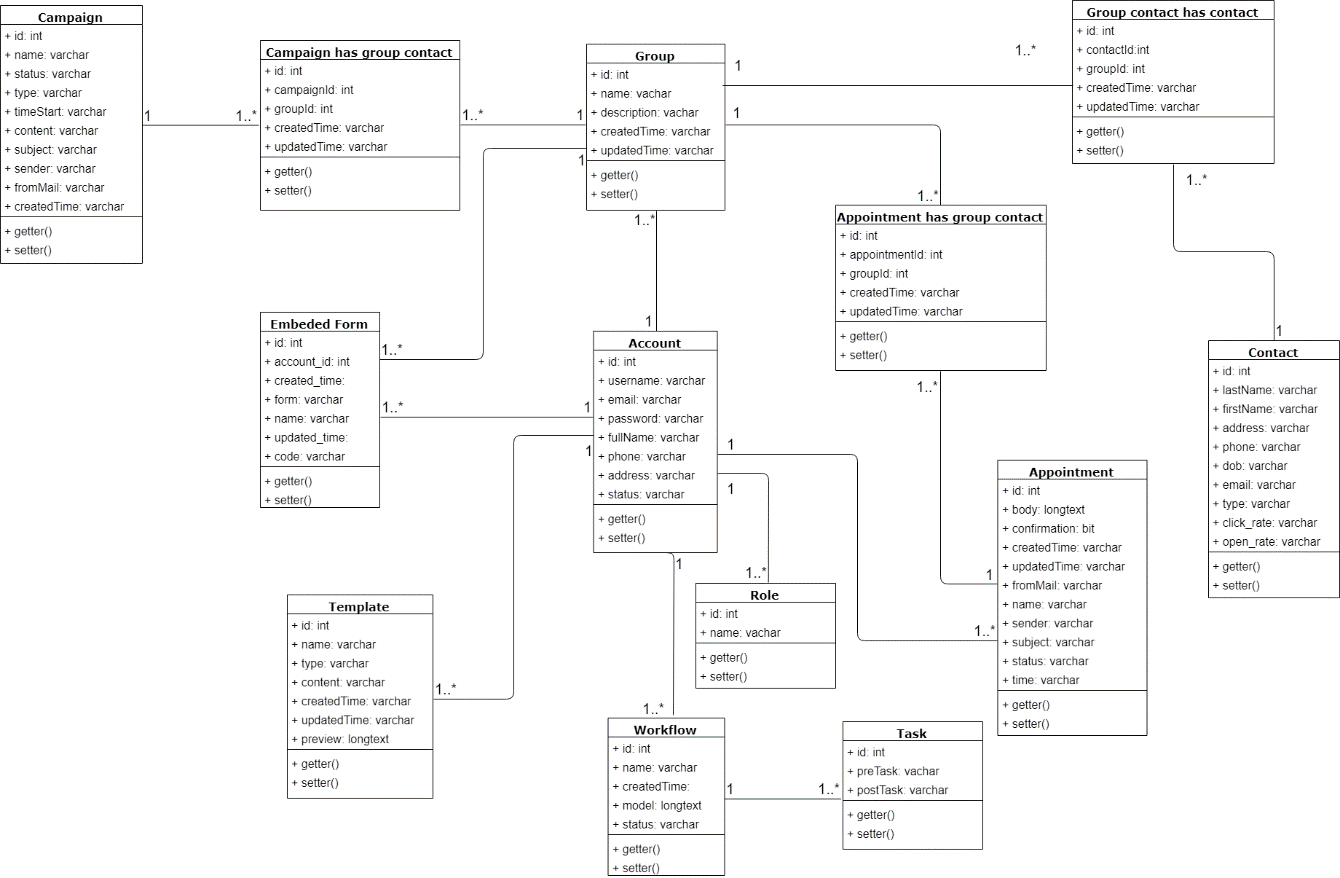
**Figure Component Diagram**

|  |  |
| --- | --- |
| **Component dictionary** | |
| **Component name** | **Description** |
| Equipment Service | A service responsible for managing information related to equipment, hiring transaction, hiring transaction feedbacks. |
| Material Service | A service responsible for managing information related to materials, material transactions, material transaction feedbacks. |
| Debris Service | A service responsible for managing information related to debris, debris post, debris bids, debris transactions, debris transaction feedbacks. |
| Registered User Service | A service responsible for managing information related to Registered Users. |
| Construction Service | A service responsible for managing information related to constructions. |
| Authentication Service | A service responsible for authentication and authorization. |
| Image Storage Service | A service responsible for uploading images, including equipment images, material images, debris images, profile images. |
| Scheduler Service | A service responsible for time-related tasks. |
| Notification Service | A service responsible for managing notification |
| Repository | A service responsible for data management. |
| Elastic Search Service | A provided service of Elastic search, responsible for low-latency performance information gathering and managing searchable resource |
| Device Messaging Service | An external service responsible for send notifications to actual devices. This service can be realized by Firebase Cloud Messaging service and Expo Messaging service. |
| File Storage Service | An external service responsible for raw resource storage. This service can be realized by Google Cloud Storage service, Imgur Storage service, etc. |
| Price Suggestion Service | A service responsible for model training, data collecting data, data processing, model applying for price suggestion. |

Table Component Data Dictionary

## Detailed Description

### Class Diagram



### Class Diagram Explanation

|  |  |  |
| --- | --- | --- |
| **CLASS DICTIONARY: DESCRIBE CLASS** | | |
| ***Class Name*** | ***Mapping column with Conceptual diagram*** | ***Description*** |
| **Account** | Account | Contains the account information |
| **Role** | Role | Contains the account role |
| **Group** | Group | Contains the group information |
| **Contact** | Contact | Contains the contact information |
| **Group contact has contact** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the group’s contact information |
| **Campaign** | Campaign | Contains the campaign information |
| **Campaign has group contact** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the campaign’s group contact information |
| **Appointment** | Appointment | Contains the appointment information |
| **Appointment has group contact** | N/A | Not exist in conceptual diagram, but needed in class diagram to contain the appointment’s group contact information |
| **Embedded Form** | Embeded Form | Contains the embedded form information |
| **Template** | Template | Contains the template information |
| **Workflow** | Workflow | Contains the workflow information |
| **Task** | Task | Contains the workflow’s task information |

Table Class Diagram dictionary

#### Account

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of account |
| username | string | private | Username of account |
| password | string | private | Password of account |
| fullName | string | private | Full name of account |
| phone | string | private | Phone number of account |
| email | string | private | Email of account |
| address | string | private | Address of account |
| status | string | private | Status of account |

Table - <Class Diagram> Account Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get account attributes |
| setter() | void | public | Set value of account attributes |

Table - <Class Diagram> Account Methods

#### Role

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of role |
| name | string | private | role of account |

Table - <Class Diagram> Role Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get role attributes |
| setter() | void | public | Set value of role attributes |

Table - <Class Diagram> Role Methods

#### Group

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of group |
| name | string | private | name of group |
| description | string | private | Description of group |
| createdTime | string | private | Created time of group |
| updatedTime | string | private | Latest updated time of group |

Table - <Class Diagram> Group Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get group attributes |
| setter() | void | public | Set value of group attributes |

Table - <Class Diagram> Group Methods

#### Contact

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of contact |
| lastName | string | private | Last name of contact |
| firstName | string | private | First name of contact |
| address | string | private | Address of contact |
| phone | string | private | Phone number of contact |
| dob | string | private | Date of birth of contact |
| email | string | private | email of contact |
| type | string | private | Type of contact |
| click\_rate | string | private | Clicked rate of contact |
| open\_rate | string | private | Opened rate of contact |

Table - <Class Diagram>Contact Atributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get contact attributes |
| setter() | void | public | Set value of contact attributes |

Table - <Class Diagram>Contact Methods

#### Group contact has contact

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of group contact has contact |
| contactId | int | private | Contact ID of group contact has contact |
| groupId | int | private | Group ID of group contact has contact |
| createdTime | string | private | Created time of group contact has contact |
| updatedTime | string | private | Latest updated time of group contact has contact |

Table - <Class Diagram>Group contact has contact Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get group contact has contact attributes |
| setter() | void | public | Set value of group contact has contact attributes |

Table - <Class Diagram> Group contact has contact Methods

#### Campaign

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of campaign |
| name | string | private | Name of campaign |
| status | string | private | Status of campaign |
| createdTime | string | private | Created time of campaign |
| type | string | private | Type of campaign |
| content | string | private | Content of campaign |
| subject | string | private | Subject mail of campaign |
| sender | string | private | Sender name of campaign |
| fromMail | string | private | From mail of campaign |

Table - <Class Diagram>Campaign Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get campaign attributes |
| setter() | void | public | Set value of campaign attributes |

Table - <Class Diagram> Campaign Methods

#### Campaign has group contact

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of campaign has group contact |
| campaignId | int | private | Campaign Id of campaign has group contact |
| groupId | int | private | Group ID of campaign has group contact |
| createdTime | string | private | Created time of campaign has group contact |
| updatedTime | string | private | Updated time of campaign has group contact |

Table - <Class Diagram>Campaign has group contact Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get campaign has group contact attributes |
| setter() | void | public | Set value of campaign has group contact attributes |

Table - <Class Diagram> Campaign has group contact Methods

#### Appointment

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of appointment |
| name | string | private | Name of appointment |
| status | string | private | Status of appointment |
| createdTime | string | private | Created time of appointment |
| type | string | private | Type of appointment |
| body | string | private | body of appointment |
| subject | string | private | Subject mail of appointment |
| sender | string | private | Sender name of appointment |
| fromMail | string | private | From mail of appointment |
| confirmation | boolean | private | Clicked status of appointment |
| time | string | private | Hole time of appointment |

Table - <Class Diagram>Appointment Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get appointment attributes |
| setter() | void | public | Set value of appointment attributes |

Table - <Class Diagram> Appointment Methods

#### Appointment has group contact

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of appointment has group contact |
| appointmentId | int | private | Campaign Id of appointment has group contact |
| groupId | int | private | Group ID of appointment has group contact |
| createdTime | string | private | Created time of appointment has group contact |
| updatedTime | string | private | Updated time of appointment has group contact |

Table - <Class Diagram>Appointment has group contact Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get appointment has group contact attributes |
| setter() | void | public | Set value of appointment has group contact attributes |

Table - <Class Diagram> Appointment has group contact Methods

#### Embedded Form

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of embedded form |
| name | string | private | Name of embedded form |
| form | string | private | Form design of embedded form |
| createdTime | string | private | Created time of embedded form |
| updatedTime | string | private | Updated time of embedded form |

Table - <Class Diagram>Embedded Form Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get embedded form attributes |
| setter() | void | public | Set value of embedded form attributes |

Table - <Class Diagram> Embedded Form Methods

#### Template

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of template |
| name | string | private | Name of template |
| type | string | private | Type of template |
| content | string | private | Content of template |
| preview | string | private | Image preview link of template |
| createdTime | string | private | Created time of template |
| updatedTime | string | private | Updated time of template |

Table - <Class Diagram>Template Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get template attributes |
| setter() | void | public | Set value of template attributes |

Table - <Class Diagram> Template Methods

#### Workflow

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of workflow |
| name | string | private | Name of workflow |
| model | string | private | Model of workflow |
| status | string | private | Status of workflow |
| createdTime | string | private | Created time of workflow |

Table - <Class Diagram>Workflow Attributes

* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get workflow attributes |
| setter() | void | public | Set value of workflow attributes |

Table - <Class Diagram> Workflow Methods

#### 4.2.13. Task

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique identifier of task |
| preTask | string | private | Previous task of task |
| postTask | string | private | Post task of task |

Table - <Class Diagram>Task Attributes

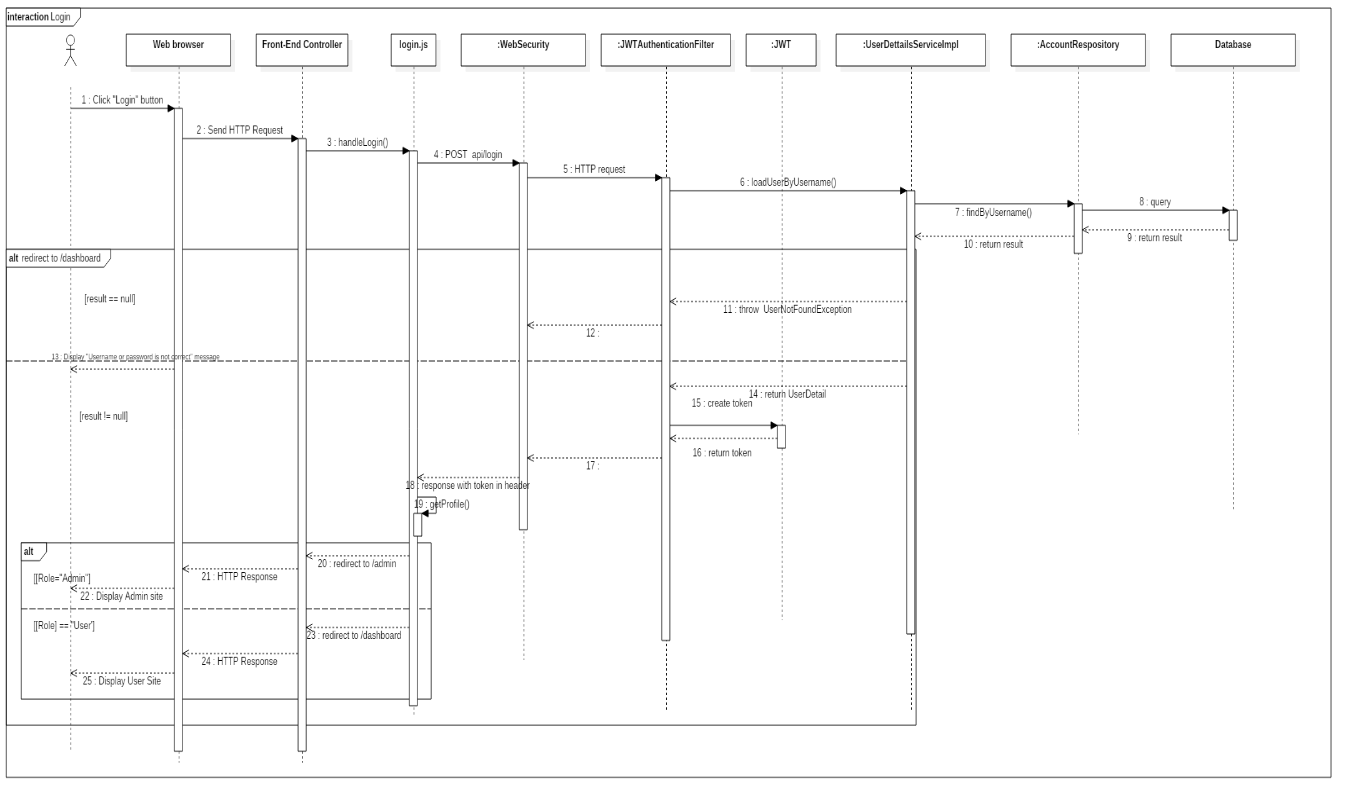
* **Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter() | attribute type | public | Get task attributes |
| setter() | void | public | Set value of task attributes |

Table - <Class Diagram> Task Methods

### Interaction Diagram

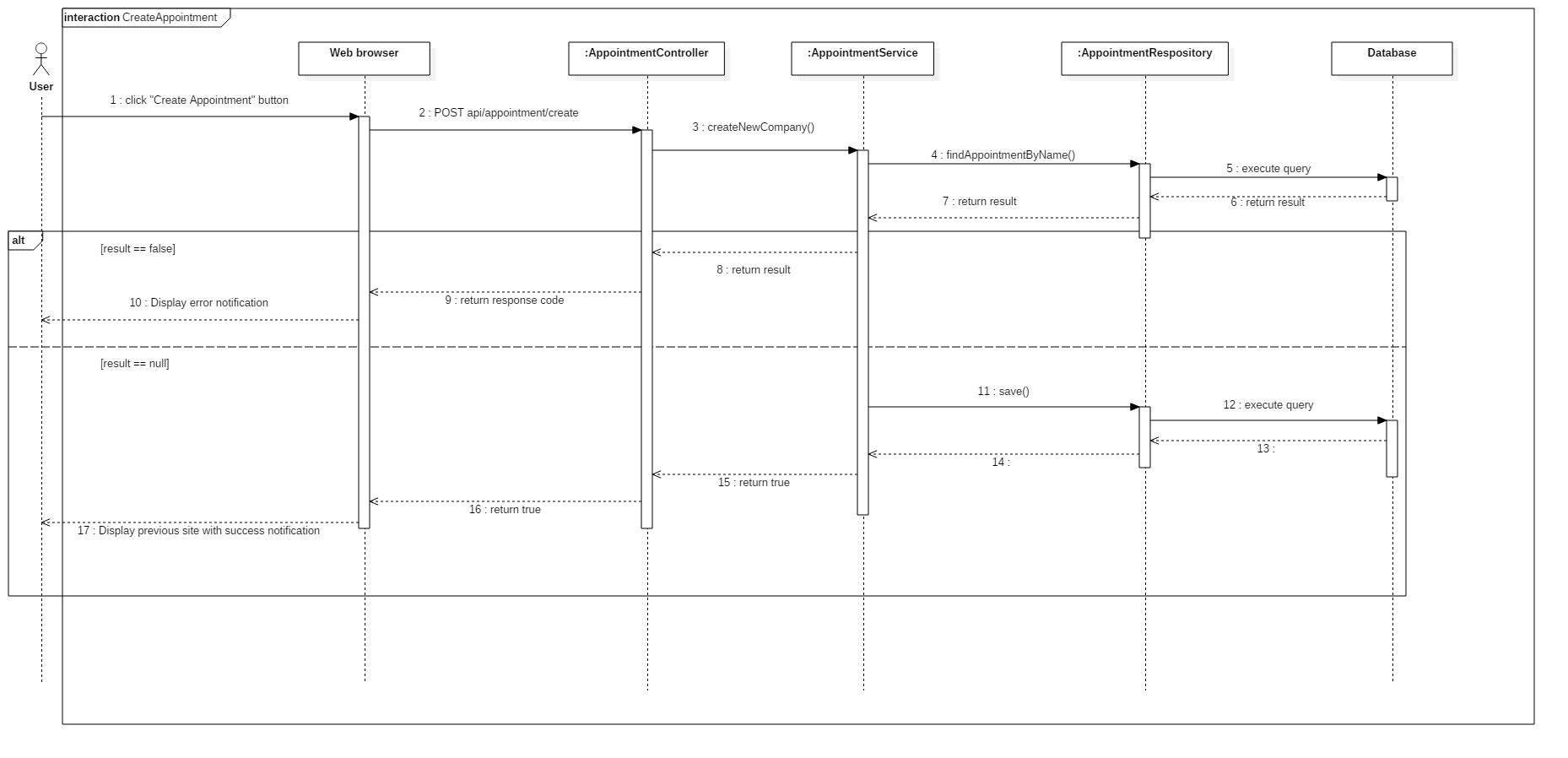
#### 4.3.1 Login



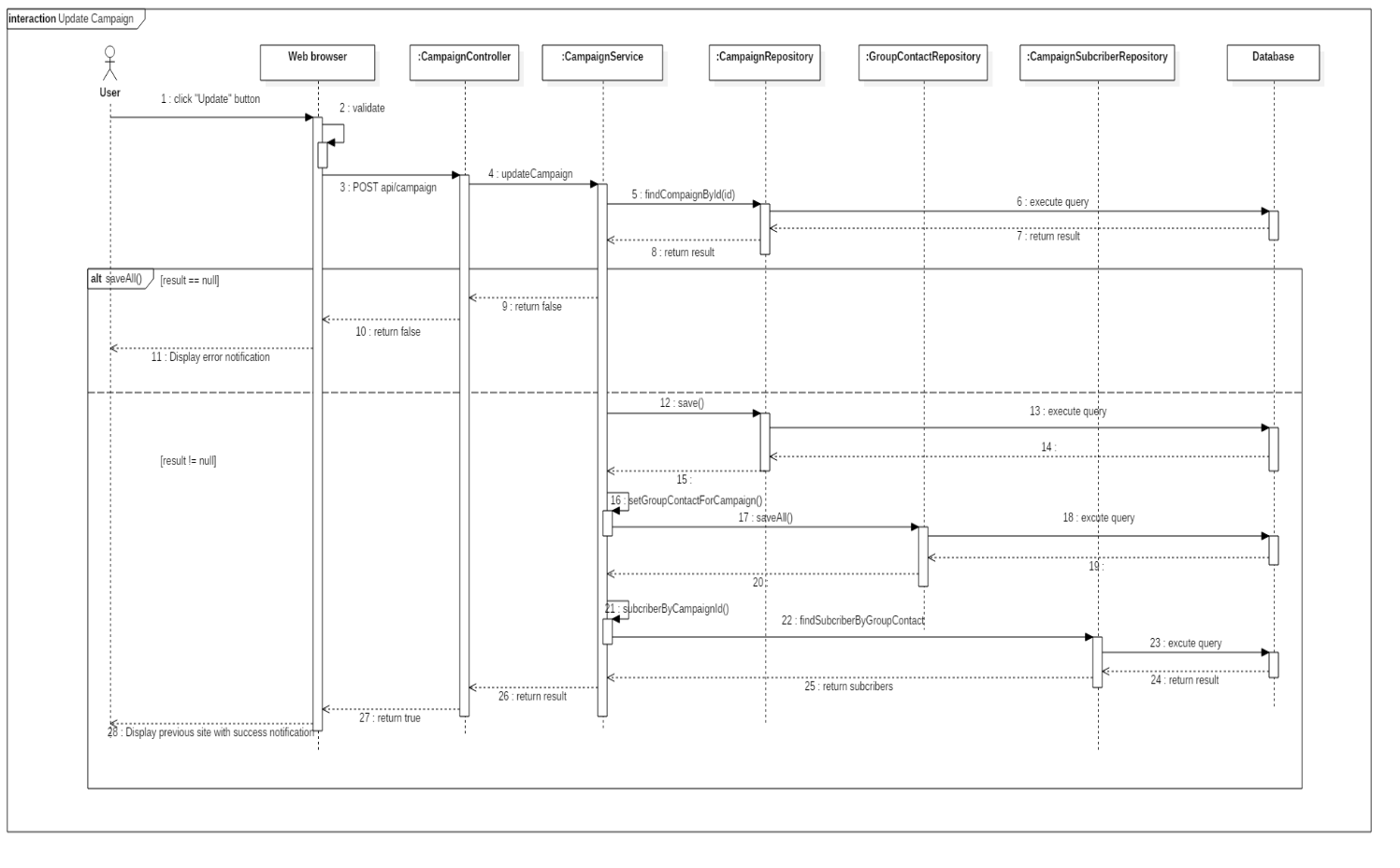
#### Create Campaign



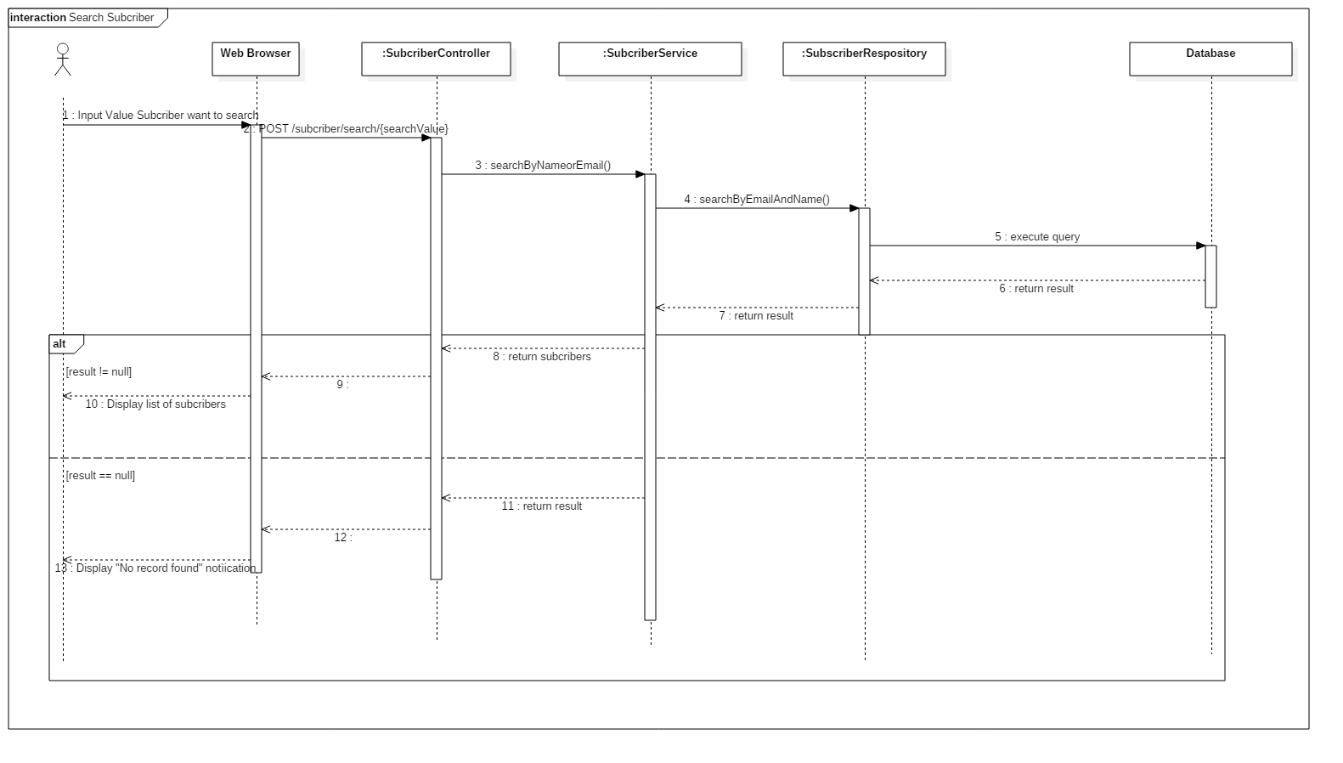
#### Create Appointment



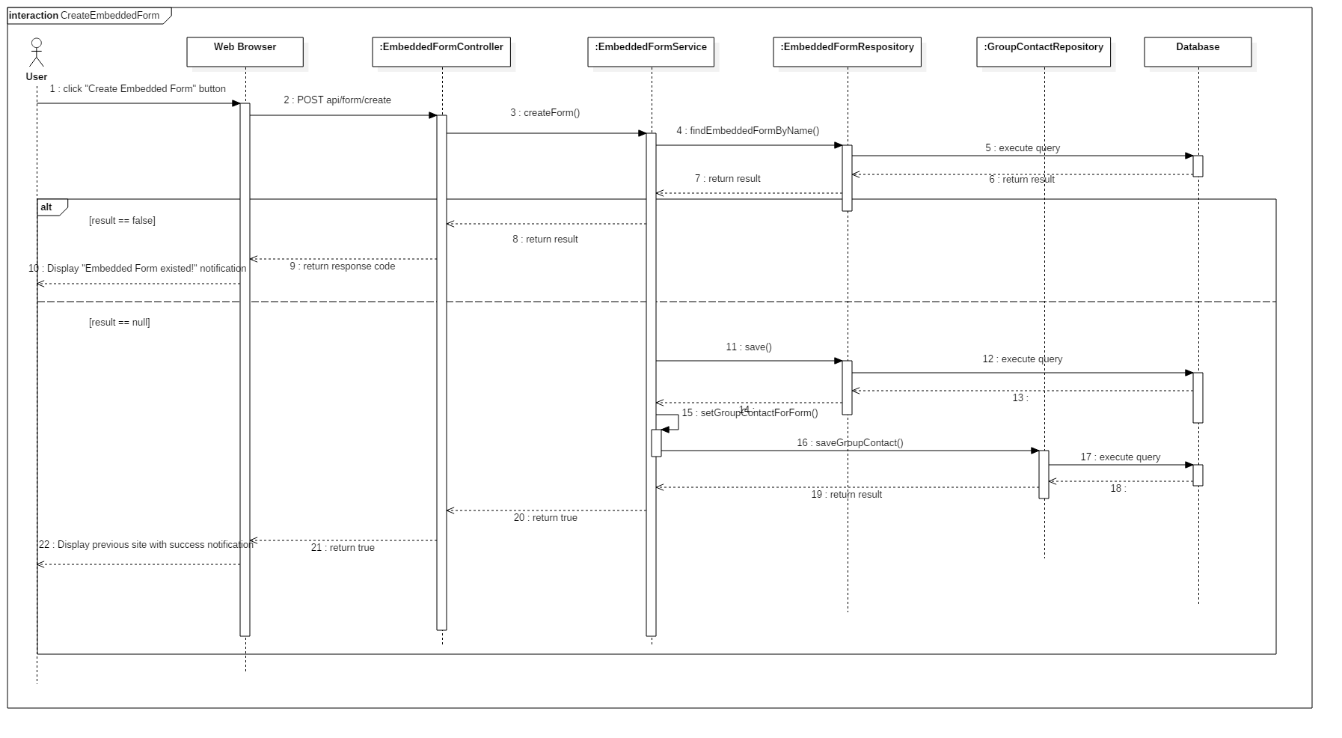
#### Update Campaign



#### Search Subscriber



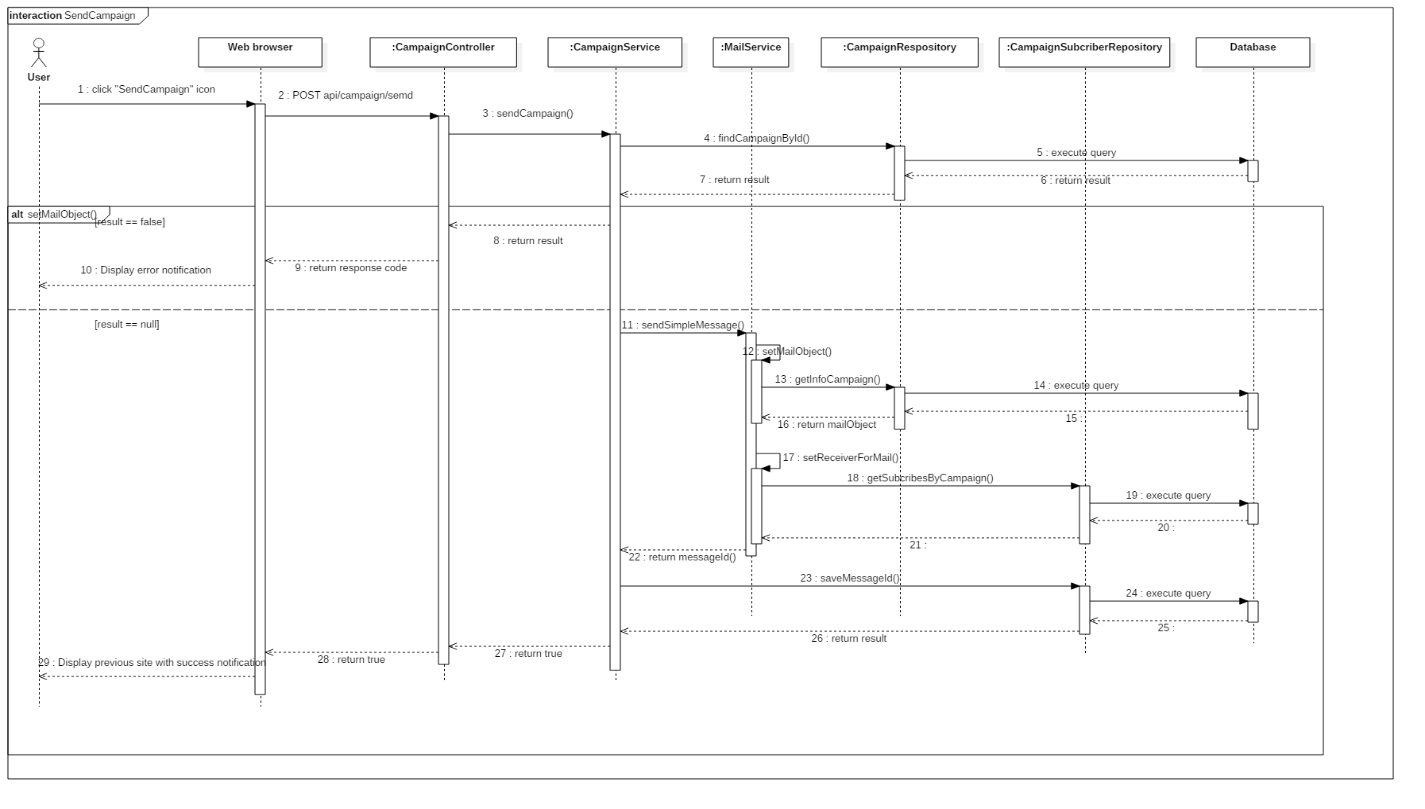
#### Create Embedded Form



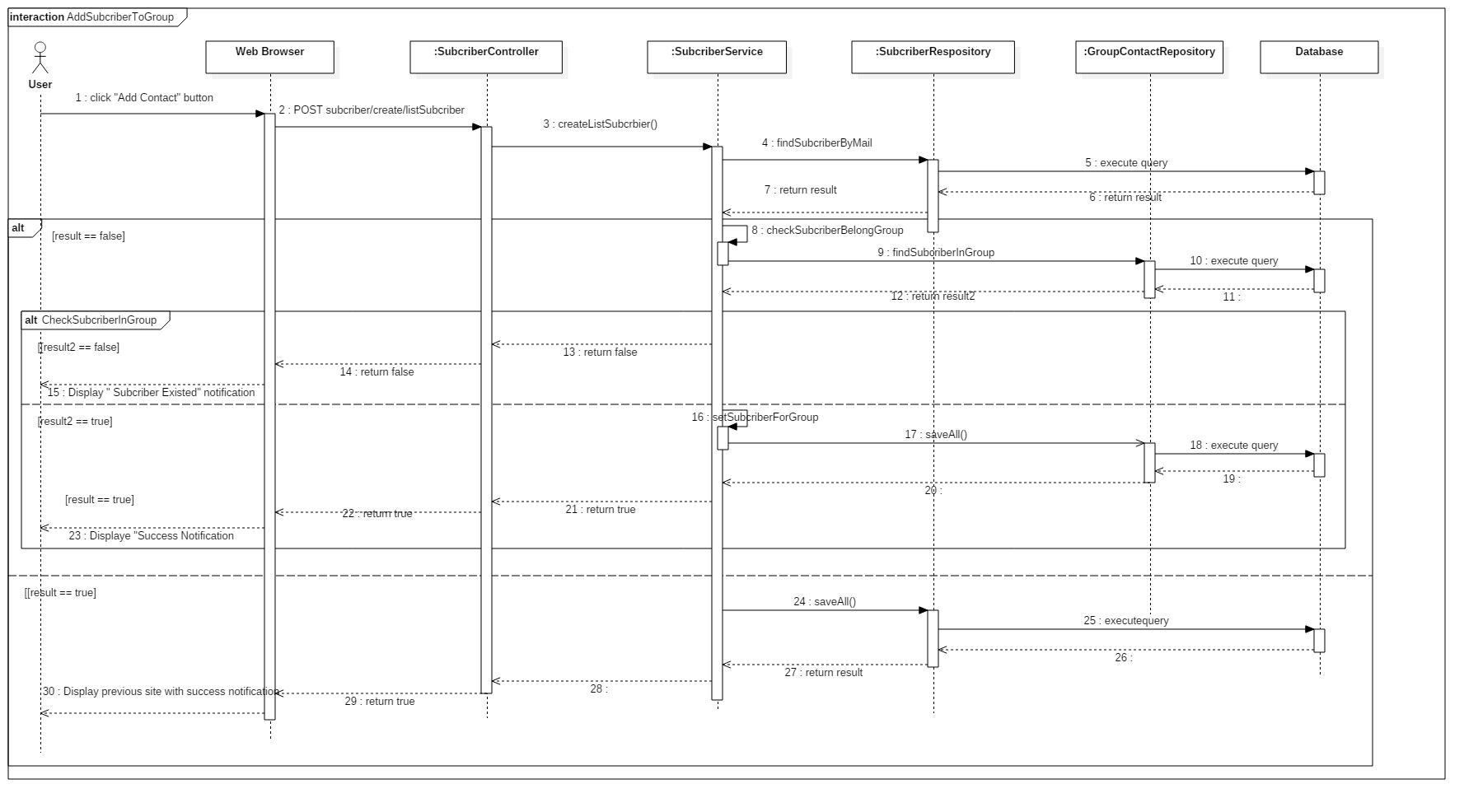
#### Create Workflow



#### Send Campaign



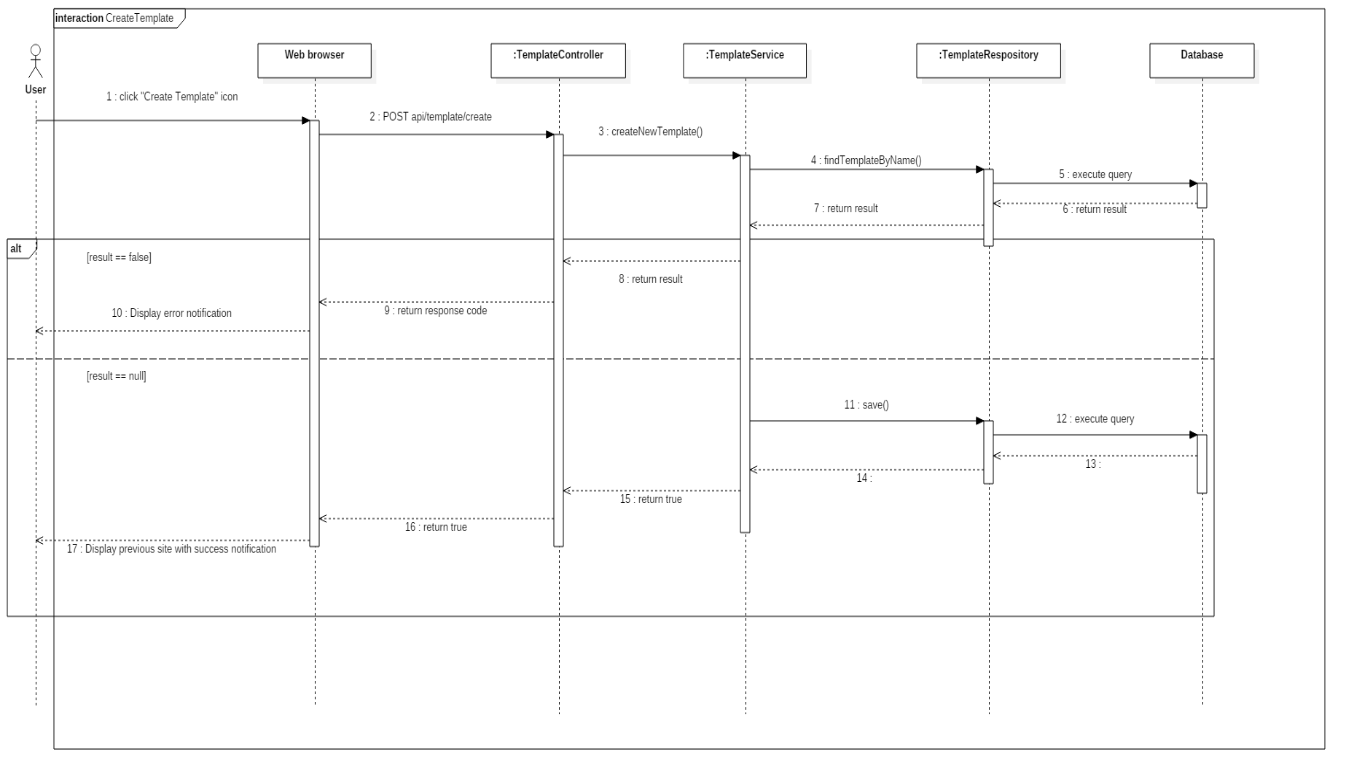
#### Add Subcriber To Group



#### Pause Workflow



#### Create Template



## Interface

### Component Interface

### User Interface Design

#### Registered User web application

##### Login



Figure 23 Login

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Username | Username to login | No | Yes | TextBox | String |
| 2 | Password | Password to login | No | Yes | TexBox | String |

Table <Fields> Login

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Login | Login into the system | Required | User logged in into the system |
| 4 | Forgot Password | User forget and want to take back password | N/A | User is redirected to a page for filling in new password |
| 5 | [Not a member? SIGN UP](http://localhost:3000/register) | User sign up new account | N/A | User is redirected into the sign up page |

Table <Buttons/Hyperlinks> Login

##### Sign Up



**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Username | Username to sign up | No | Yes | TextBox | String |
| 2 | Email | Email to sign up | No | Yes | Textbox | String |
| 3 | Password | Password to sign up | No | Yes | TexBox | String |
| 4 | Re-Confirm password | Confirm enterred password | No | Yes | TextBox | String |

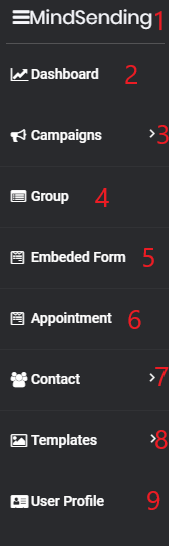
Table <Fields> Sign Up

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Register | User sign up new account | Required | User signed up new account successfully |

Table <Buttons/Hyperlinks> Sign Up

##### Main menu

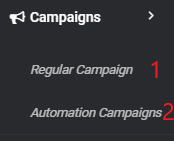


**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | MindSending | Home | User must had been authorized | Redirect to home page |
| 2 | Dashboard | Report page | User must had been authorized | Redirect to report page |
| 3 | Campaign | Campaign page | User must had been authorized | Redirect to campaign page |
| 4 | Group | Group contact page | User must had been authorized | Redirect to group contact page |
| 5 | Embed Form | Embed form page | User must had been authorized | Redirect to embed form page |
| 6 | Appointment | Appointment page | User must had been authorized | Redirect to appointment page |
| 7 | Contact | Contact page | User must had been authorized | Redirect to contact page |
| 8 | Template | Template page | User must had been authorized | Redirect to template page |
| 9 | User profile | User profile page | User must had been authorized | Redirect to user profile page |

Table <Buttons/Hyperlinks> Main menu

##### Campaign menu



**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Regular Campaign | Regular Campaign page | User must had been authorized | Redirect to Regular Campaign page |
| 2 | Automation Campaigns | Automation Campaigns page | User must had been authorized | Redirect to Automation Campaigns page |

Table <Buttons/Hyperlinks> Campaign Menu



**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Find materials | Search group by name | N/A | Group with corresponding result |
| 2 | Create Group | Button for creating group | User must had been authorized | Redirect to create group pop-up |

Table <Buttons/Hyperlinks> Group

##### Embed form



**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Search Form | Search form by name | N/A | Form with corresponding result |
| 2 | Create Form | Button for creating form | User must had been authorized | Redirect to create form page |

Table <Buttons/Hyperlinks> Embed form

##### Appointment



**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Search Appointment | Search appointment by name | N/A | appointment with corresponding result |
| 2 | Create Invite Mail | Button for creating appointment | User must had been authorized | Redirect to create appointment page |

Table <Buttons/Hyperlinks> appointment

##### Contact

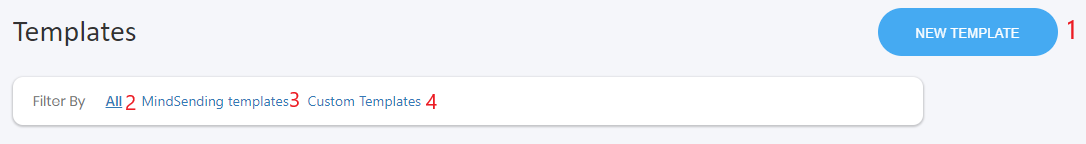


**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Search Contact | Search contact by name | N/A | Contact with corresponding result |
| 2 | Create Contacts | Button for creating Contact | User must had been authorized | Redirect to create Contacts page |

Table <Fields> Equipment search

##### Templates



**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | New Template | Button for creating template | User must had been authorized | Redirect to create template page |
| 2 | All | List all template | N/A | All templates are listed |
| 3 | MindSending templates | List all MindSending templates | N/A | All MindSending templates are listed |
| 4 | Custom templates | List all Custom templates | N/A | All Custom templates are listed |

Table <Buttons/Hyperlinks> template

## Database Design

### Entity Relationship Diagram (ERD)

Figure 24 Entity Relation Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe content of all entities** | |
| **Entity name** | **Description** |
| Account | Contains the user’s information. |
| Contact | Contains the contact’s information. |
| Campaign | Contains the campaign’s information. |
| Template | Contains the template’s information. |
| Appointment | Contains the appointment’s information. |
| Embed form | Contains the embed form’s information. |
| Group | Contains the group’s information. |
| Workflow | Contains the workflow’s information. |
| Task | Contains the task’s information. |
| Role | Contains the role’s information. |

Table Entity Relationship Diagram Data dictionary

## Algorithms EMP

### Adjust the level of contact with the result returned of amazon service by using linear programming algorithm

#### Definition

This algorithm is intended to adjust level of contact that are counted by the result returned of amazon service.

#### Define problem

Help user to identify exactly the engagement of a contact so that they will have suitable strategy later.

#### Solution

|  |
| --- |
|  |

Based on user criteria, we have the following general formula:

With:

f(x): point that will be plus for that user

a(x): point that will be plus after a campaign

b(x): point that will be plus for the current user’s level

*Based on the cumulative score, there are the following levels:*

* + Bronze: 30 exp
  + Silver : 65 exp
  + Gold : 102 exp
  + Platinum : 247 exp
  + Diamond : 534 exp

Based on user criteria, we have the following point will be plus after a campaign-a(x) formula:

**Supposed:**

O(x): Number of Emails that user Opened

C(x): Number of Emails that user Clicked

D(x): Number of Emails that user receive

P(o): Open rate

P(c): Click rate

**We have:**

P(o) =

P(c) =

* + If P(o) applied: a(x) =
  + **Plus 5 point**
  + If P(o) applied: a(x) =
  + **Plus 10 point**
  + If P(o) applied: a(x) =
  + **Plus 15 point**
  + If P(c) applies: a(x) =
  + **Plus 10 point**
  + If P(c) applies: a(x) =
  + **Plus 15 point**
  + If P(c) applies: a(x) =
  + **Plus 20 point**
  + If: D(x) = 1
  + **Plus 2 point**
  + If: O(x) > 3 in a row
  + **Plus 10 point**
  + If: C(x) > 3 in a row
  + **Plus 20 point**
  + If: User do not open email more than 3 times
  + **Minus 5 point**

\* Points are only added at the end of the day

Based on user criteria, we have the following point will be plus for the current user’s level-b(x) formula:

**Supposed:**

L: current user’s level

**We have:**

b = (L +1) \* ln(L)

**For example:**

The current level of that user is 2

* + b = (2+1) \* ln(2)

#### Complexity

In total, the complexity of this algorithm is **O ().**

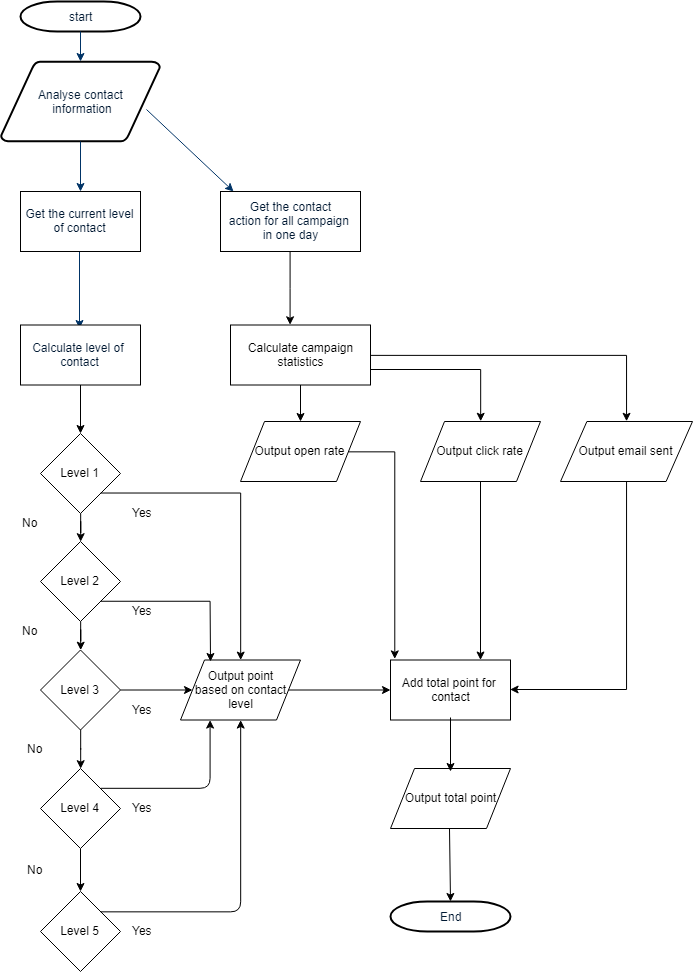
#### Example

Example:

We have a table of contacts with their current level and the point that will be added for each level:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Contact level** | Bronze | Silver | Gold | Platinum | Diamond |
| **Contact name** | John | Joseph | David | Ben | Tony |
| **Mails are sent** | 4 | 4 | 5 | 5 | 6 |
| **Open rate in a day** | 75% | 25% | 40% | 20% | 66.67% |
| **Click rate in a day** | 50% | 25% | 20% | 0% | 33.33% |
| **Point for each level in a day** | 0 | 2 | 4 | 7 | 10 |
| **Total point** | 33 | 25 | 39 | 22 | 47 |

#### Flow chart



# System implementation & Test

## Introduction

### Overview

This section provides in detail all necessary information about implementation information and testing procedure of EMP includes test plans, test cases, test result and risks estimations.

### Test Approach

* Goal: To test the whole system based on the core workflow.
* Method: System Testing, Black-box Testing.

## Database Relationship Diagram

### Physical Diagram

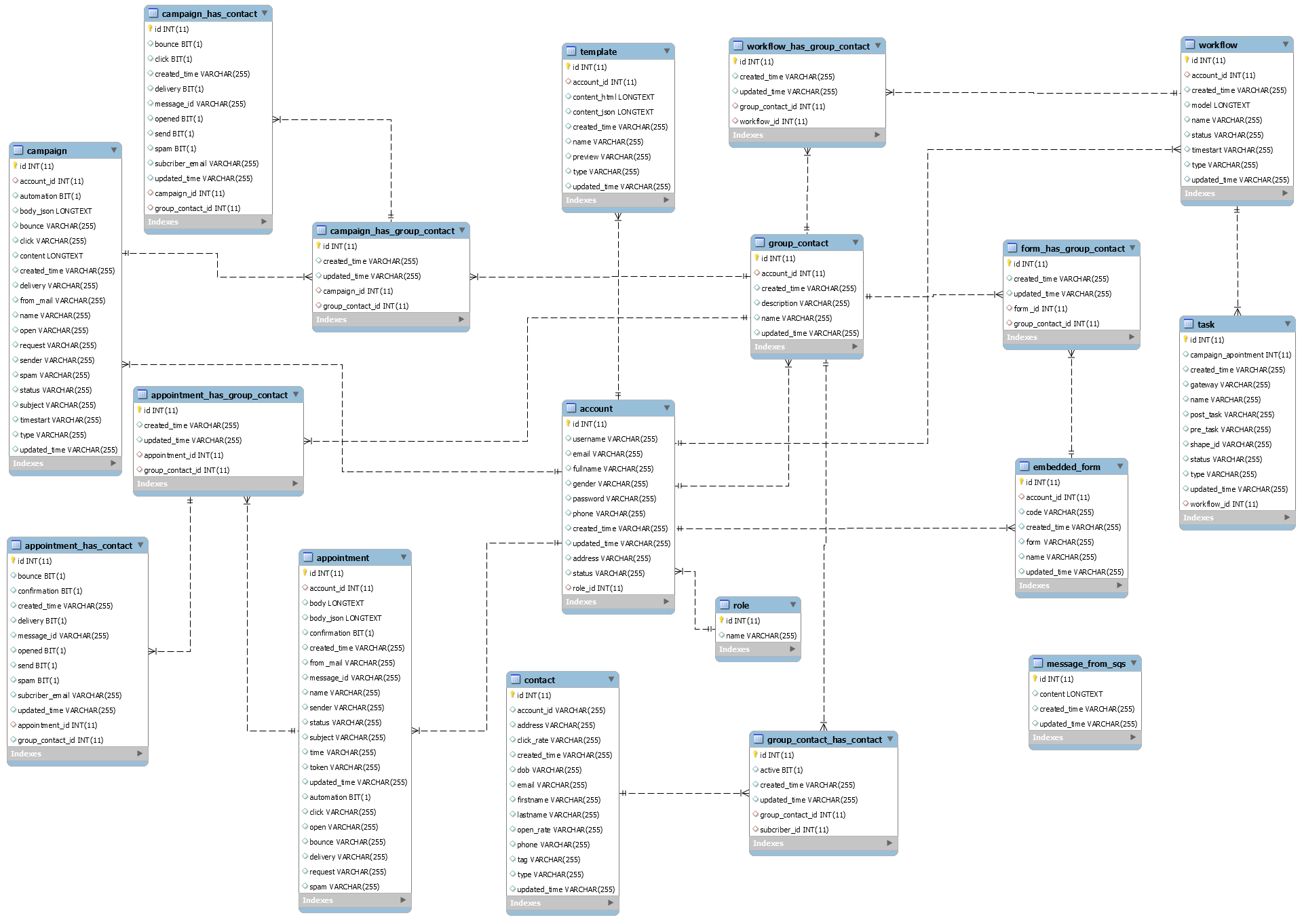


Figure 27: Physical diagram

### Data dictionary

|  |  |
| --- | --- |
| ***Entity Name*** | ***Description*** |
| **account** | Describe all account in the system |
| **role** | Describe all role of account in the system |
| **group\_contact** | Describe all group of account in the system |
| **contact** | Describe all contact of account in the system |
| **group\_contact\_has\_contact** | Describe all role of account in the system |
| **campaign** | Describe all campaign of account in the system |
| **campaign\_has\_group\_contact** | Describe all group of campaign of account in the system |
| **campaign\_has\_contact** | Describe all contact of campaign of account in the system |
| **appointment** | Describe all appointment of account in the system |
| **appointment\_has\_group\_contact** | Describe all group of appointment of account in the system |
| **appointment\_has\_contact** | Describe all contact of appointment of account in the system |
| **embedded\_form** | Describe all embedded form of account in the system |
| **form\_has\_group\_contact** | Describe all group of embedded form of account in the system |
| **template** | Describe all template of account in the system |
| **workflow** | Describe all workflow of account in the system |
| **task** | Describe all task of workflow of account in the system |
| **workflow\_has\_group\_contact** | Describe all group contact of workflow of account in the system |
| **message\_from\_sqs** | Describe all message detail of email from sqs of account in the system |

Table Physical diagram data dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Nulls |
| account | id | ID of user | int | No |
| username | Username of user | varchar(255) | No |
| password | Password of user | varchar(255) | No |
| email | Email of user | varchar(255) | Yes |
| phone | Phone of user | varchar(255) | Yes |
| address | Address of user | varchar(255) | Yes |
| fullname | real name of user | varchar(255) | Yes |
| contact | id | ID of contact | int | No |
| lastname | Last name of contact | varchar(255) | No |
| firstname | First name of contact | varchar(255) | No |
| address | Address of contact | varchar(255) | Yes |
| phone | Phone of contact | varchar(255) | Yes |
| email | Email of contact | varchar(255) | Yes |
| dob | Birthday of contact | varchar(255) | Yes |
| type | Type of contact | varchar(255) | No |
| click\_rate | Click rate of contact | varchar(255) | No |
| open\_rate | Open rate of contact | varchar(255) | No |
| campaign | id | ID of campaign | int | No |
| name | Name of campaign | varchar(255) | No |
| status | Status of campaign | varchar(255) | No |
| type | Type of campaign | varchar(255) | No |
| timestart | Time start of campaign | varchar(255) | No |
| content | Content of campaign | varchar(255) | No |
| subject | Subject of campaign | varchar(255) | No |
| sender | Sender of campaign | varchar(255) | No |
| fromMail | From mail of campaign | varchar(255) | No |
| createdTime | Created time of campaign | varchar(255) | No |
| template | id | ID template | int | No |
| name | Name of template | varchar(255) | No |
| type | Type of template | varchar(255) | Yes |
| content | Content of template | varchar(255) | Yes |
| createdTime | Created time of template | varchar(255) | Yes |
| preview | Preview template | varchar(255) | Yes |
| appointment | id | ID of appointment | int | No |
| body | Body of appointment | varchar(255) | No |
| fromMail | From mail | varchar(255) | No |
| name | Name of appointment | varchar(255) | No |
| sender | Sender of appointment | varchar(255) | No |
| subject | Subject of appointment | varchar(255) | No |
| status | Status of appointment | varchar(255) | No |
| time | Time of appointment | varchar(255) | No |
| embed\_form | id | ID of embed form | int | No |
| name | ID of embed form | varchar(255) | No |
| createdTime | Created time of embed form | varchar(255) | No |
| updateTime | Update time of embed form | varchar(255) | No |
| group | id | ID of group | int | No |
| name | Name of group | varchar(255) | No |
| description | Description of group | varchar(255) | No |
| createdTime | Created time of group | varchar(255) | No |
| updatedTime | Updated time of group | varchar(255) | No |
| workflow | id | ID of workflow | int | No |
| name | Name of workflow | varchar(255) | No |
| model | Model of workflow | varchar(255) | No |
| status | Status of workflow | varchar(255) | No |
| task | id | ID of task | int | No |
| shapId | ID of shap | varchar(255) | No |
| preTask | Pre task | varchar(255) | No |
| postTask | Post task | varchar(255) | No |
| role | id | ID of role | int | No |

Table Physical data dictionary

## Test Plan

The purpose of this section is to verify and ensure that EMPSC meets its design specification and other requirements from user. The following part will describe which features to be tested and which will not.

### Features to be tested

* Registered User:
  + Create new regular campaign
  + Create new automation campaign
  + Create new group
  + Create new embedded form
  + Create new appointment
  + Create new contact
  + Create new template
  + Update profile

### Features not to be tested

* Admin:
  + Reset password
  + Reply feedback
  + View statistic

## System Testing Test Case

### Registered User testing case:

#### Create new regular campaign:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new regular campaign | 1. Click Create Campaign  2. Select option template or simple text  3. Fill nothing in textbox  4. Click “Begin” button | Begin button is disabled. Can not click to begin. | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new regular campaign | 1. Click Create Campaign  2. Select option template or simple text  3. Fill campaign name in textbox  4. Click “Begin” button | User will be redirected to design campaign page | N/A | Pass | 25/07/2019 |  |
| EMP\_03 | Registered User request for creating a new regular campaign | 1. Choose List  2. Fill in Sender Name  3. Click “Design Email” button | Design Email button is disabled. Can not click to design. | N/A | Pass | 25/07/2019 |  |
| EMP\_04 | Registered User request for creating a new regular campaign | 1. Choose List  2. Fill in Sender Name  3. Fill in Email Address  4. Click “Design Email” button | Design Email button is disabled. Can not click to design. | N/A | Pass | 25/07/2019 |  |
| EMP\_05 | Registered User request for creating a new regular campaign | 1. Choose List  2. Fill Sender Name  3. Fill in Email Address  4. Fill in Subject  5. Click “Design Email” button  6. Choose and edit template  7. Save campaign | New regular campaign is created. | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new regular campaign

#### Create new workflow:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new workflow | 1. Click Create Automation  2. Fill nothing in textbox  3. Click “Begin” button | Begin button is disabled. Can not click to begin. | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new workflow | 1. Click Create Automation  2. Fill campaign name in textbox  3. Click “Begin” button | User will be redirected to design automation campaign page | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new workflow

#### Create new group:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new group | 1. Click Create Group  2. Fill nothing in “New Name”  3. Click “Create” button | Create button is disabled. Can not click to create. | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new group | 1. Click Create Group  2. Fill in “New Name”  3. Click “Begin” button | New group will be created | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new group

#### Create new embedded form:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new embedded form | 1. Click Create Form  2. Fill nothing in “Form Name”  3. Click “Save this form” button | System shows “You must fill in name for this form”. | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new embedded form | 1. Click Create Group  2. Fill in “Form Name”  3. None of group is chosen  4. Click “Save this form” button | System shows “You must choose a group” | N/A | Pass | 25/07/2019 |  |
| EMP\_03 | Registered User request for creating a new embedded form | 1. Click Create Group  2. Fill in “Form Name”  3. Choose one group  4. Click “Save this form” button | New embedded form is created | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new embedded form

#### Create new appointment:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new appointment | 1. Click Create appointment  2. Fill nothing in appointment name  3. Click “Save draft” button | System shows “You need to enter a name that’s at least 3 character long” | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new appointment | 1. Click Create appointment  2. Fill in appointment name  3. Date and time are not chosen  4. Click “Save draft” button | System shows “Enter date and time of your appointment” | N/A | Pass | 25/07/2019 |  |
| EMP\_03 | Registered User request for creating a new appointment | 1. Click Create appointment  2. Fill in appointment name  3. Date and time are chosen  4. Groups are not chosen  5. Click “Save draft” button | System shows “Choose at least one group” | N/A | Pass | 25/07/2019 |  |
| EMP\_04 | Registered User request for creating a new appointment | 1. Click Create appointment  2. Fill in appointment name  3. Date and time are chosen  4. Groups are chosen  5. Sender name and email address are not filled in  6. Click “Save draft” button | System shows “Enter sender name and email address” | N/A | Pass | 25/07/2019 |  |
| EMP\_05 | Registered User request for creating a new appointment | 1. Click Create appointment  2. Fill in appointment name  3. Date and time are chosen  4. Groups are chosen  5. Sender name and email address are filled in  6. Subject is not filled in  7. Click “Save draft” button | System shows “This field can not be empty” | N/A | Pass | 25/07/2019 |  |
| EMP\_06 | Registered User request for creating a new appointment | 1. Click Create appointment  2. Fill in appointment name  3. Date and time are chosen  4. Groups are chosen  5. Sender name and email address are filled in  6. Subject is filled in  7. Click “Save draft” button | New appointment is created | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new appointment

#### Create new contact:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new contact | 1. Click Add contact  2. Choose “Manual add”  3. Choose option “add contacts” or “add contacts and include in an existing list”  4. Fill nothing in email field  5. Click “Save Contact” button | System shows “Please enter email address” | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new contact | 1. Click Add contact  2. Choose “Manual add”  3. Choose option “add contacts” or “add contacts and include in an existing list”  4. Fill in email field  5. Click “Save Contact” button | New contacts are created | N/A | Pass | 25/07/2019 |  |
| EMP\_03 | Registered User request for creating a new contact | 1. Click Add contact  2. Choose “Upload CSV”  3. Choose option “add contacts” or “add contacts and include in an existing list”  4. CSV file is not chosen  5. Click “Save Contact” button | System shows “Please choose a CSV file” | N/A | Pass | 25/07/2019 |  |
| EMP\_04 | Registered User request for creating a new contact | 1. Click Add contact  2. Choose “Upload CSV”  3. Choose option “add contacts” or “add contacts and include in an existing list”  4. CSV file is chosen  5. Click “Save Contact” button | New contacts are created | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new contact

#### Create new template:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for creating a new template | 1. Click New Template  2. Drag and drop nothing component  3. Click “Save template” button | System shows “You must drag and drop at least one component” | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for creating a new template | 1. Click New Template  2. Drag and drop component  3. Fill nothing in template name  4. Click “Save template” button | System shows “Please enter a name for this template” | N/A | Pass | 25/07/2019 |  |
| EMP\_03 | Registered User request for creating a new template | 1. Click New Template  2. Drag and drop component  3. Fill in template name  4. Click “Save template” button | A new template is created | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create new template

#### Update profile:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | Registered User request for updating profile | 1. Click User profile  2. Edit nothing  3. Click “Update profile” button | System shows “Nothing has been changed” | N/A | Pass | 25/07/2019 |  |
| EMP\_02 | Registered User request for updating profile | 1. Click User profile  2. Edit one or more field  3. Click “Update profile” button | System shows “User profile has been updated” | N/A | Pass | 25/07/2019 |  |

Table <Test Case> Create profile

* + 1. テストケース

#### 新しい通常のキャンペーンを作成する:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 新しい通常のキャンペーンを作成するための登録ユーザーのリクエスト | 1. [キャンペーンの作成]をクリックします  2.オプションテンプレートまたは単純なテキストを選択します  3.テキストボックスに何も記入しない  4.「開始」ボタンをクリックします | 開始ボタンは無効です。クリックして開始することはできません。 | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 新しい通常のキャンペーンを作成するための登録ユーザーのリクエスト | 1. [キャンペーンの作成]をクリックします  2.オプションテンプレートまたは単純なテキストを選択します  3.テキストボックスにキャンペーン名を入力します  4.「開始」ボタンをクリックします | ユーザーはデザインキャンペーンページにリダイレクトされます | N/A | パス | 25/07/2019 |  |
| EMP\_03 | 新しい通常のキャンペーンを作成するための登録ユーザーのリクエスト | 1.リストを選択  2.送信者名を入力します  3. [メールのデザイン]ボタンをクリックします | [デザインメール] ボタンは無効になっています。クリックして設計することはできません。 | N/A | パス | 25/07/2019 |  |
| EMP\_04 | 新しい通常のキャンペーンを作成するための登録ユーザーのリクエスト | 1.リストを選択  2.送信者名を入力します  3.メールアドレスを入力してください  4. [メールのデザイン]ボタンをクリックします | [デザインメール] ボタンは無効になっています。クリックして設計することはできません。 | N/A | パス | 25/07/2019 |  |
| EMP\_05 | 新しい通常のキャンペーンを作成するための登録ユーザーのリクエスト | 1.リストを選択  2.送信者名の入力  3.メールアドレスを入力してください  4.件名を記入  5.「メールのデザイン」ボタンをクリックします  6.テンプレートを選択して編集する  7.キャンペーンを保存 | 新しい通常のキャンペーンが作成されます。 | N/A | パス | 25/07/2019 |  |

Table <Test Case> 新しい通常のキャンペーンを作成する

#### 新しいグループを作成する:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 新しいグループを作成するための登録ユーザーのリクエスト | 1. [グループの作成]をクリックします  2.「新しい名前」には何も記入しません  3.「作成」ボタンをクリックします | [作成] ボタンは無効です。クリックして作成できません。 | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 新しいグループを作成するための登録ユーザーのリクエスト | 1. [グループの作成]をクリックします  2.「新しい名前」を入力します  3.「開始」ボタンをクリックします | 新しいグループが作成されます | N/A | パス | 25/07/2019 |  |

Table <Test Case> 新しいグループを作成する

#### 新しい自動化キャンペーンを作成する:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 新しい自動化キャンペーンを作成するための登録ユーザーのリクエスト | 1. [オートメーションの作成]をクリックします  2.テキストボックスに何も記入しない  3.「開始」ボタンをクリックします | 開始ボタンは無効です。クリックして開始することはできません。 | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 新しい自動化キャンペーンを作成するための登録ユーザーのリクエスト | 1. [オートメーションの作成]をクリックします  2.テキストボックスにキャンペーン名を入力します  3.「開始」ボタンをクリックします | ユーザーはデザイン自動化キャンペーンページにリダイレクトされます | N/A | パス | 25/07/2019 |  |

Table <Test Case> 新しい自動化キャンペーンを作成する

#### 新しい埋め込みフォームの作成:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 新しい埋め込みフォームを作成するための登録ユーザーのリクエスト | 1.フォームの作成をクリックします  2.「フォーム名」に何も記入しない  3.「このフォームを保存」ボタンをクリックします | システムには「このフォームの名前を入力する必要があります」と表示されます。 | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 新しい埋め込みフォームを作成するための登録ユーザーのリクエスト | 1. [グループの作成]をクリックします  2.「フォーム名」を入力します  3.グループが選択されていません  4. [このフォームを保存]ボタンをクリックします | システムに「グループを選択する必要があります」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_03 | 新しい埋め込みフォームを作成するための登録ユーザーのリクエスト | 1. [グループの作成]をクリックします  2.「フォーム名」を入力します  3. 1つのグループを選択します  4. [このフォームを保存]ボタンをクリックします | 新しい埋め込みフォームが作成されます | N/A | パス | 25/07/2019 |  |

Table <Test Case> 新しい埋め込みフォームの作成

#### 4.2.5. 新しい予定の作成:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 登録済みユーザーが新しい予定を作成するためのリクエスト | 1. [予定を作成]をクリックします  2.予約名に何も記入しない  3.「下書きを保存」ボタンをクリックします | システムに「少なくとも3文字以上の名前を入力する必要があります」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 登録済みユーザーが新しい予定を作成するためのリクエスト | 1. [予定を作成]をクリックします  2.予定名を入力します  3.日付と時刻が選択されていません  4.「下書きを保存」ボタンをクリックします | システムに「予約の日時を入力してください」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_03 | 登録済みユーザーが新しい予定を作成するためのリクエスト | 1. [予定を作成]をクリックします  2.予定名を入力します  3.日付と時刻が選択されます  4.グループが選択されていません  5.「下書きを保存」ボタンをクリックします | システムに「少なくとも1つのグループを選択してください」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_04 | 登録済みユーザーが新しい予定を作成するためのリクエスト | 1. [予定を作成]をクリックします  2.予定名を入力します  3.日付と時刻が選択されます  4.グループが選択されます  5.送信者名とメールアドレスが入力されていません  6.「下書きを保存」ボタンをクリックします | システムに「送信者名とメールアドレスを入力してください」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_05 | 登録済みユーザーが新しい予定を作成するためのリクエスト | 1. [予定を作成]をクリックします  2.予定名を入力します  3.日付と時刻が選択されます  4.グループが選択されます  5.送信者名とメールアドレスが入力されます  6.件名は記入されていません  7.「下書きを保存」ボタンをクリックします | システムに「このフィールドは空にできません」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_06 | 登録済みユーザーが新しい予定を作成するためのリクエスト | 1. [予定を作成]をクリックします  2.予定名を入力します  3.日付と時刻が選択されます  4.グループが選択されます  5.送信者名とメールアドレスが入力されます  6.件名が記入されています  7.「下書きを保存」ボタンをクリックします | 新しい予定が作成されます | N/A | パス | 25/07/2019 |  |

Table <Test Case> 新しい予定の作成

#### 4.2.6. 新しい連絡先の作成:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 登録済みユーザーが新しい連絡先を作成するためのリクエスト | 1. [連絡先を追加]をクリックします  2.「手動追加」を選択します  3.「連絡先を追加」または「連絡先を追加して既存のリストに含める」オプションを選択します  4.メールフィールドに何も記入しない  5.「連絡先を保存」ボタンをクリックします | システムに「メールアドレスを入力してください」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 登録済みユーザーが新しい連絡先を作成するためのリクエスト | 1. [連絡先を追加]をクリックします  2.「手動追加」を選択します  3.「連絡先を追加」または「連絡先を追加して既存のリストに含める」オプションを選択します  4.メールフィールドに入力  5.「連絡先を保存」ボタンをクリックします | 新しい連絡先が作成されます | N/A | パス | 25/07/2019 |  |
| EMP\_03 | 登録済みユーザーが新しい連絡先を作成するためのリクエスト | 1. [連絡先を追加]をクリックします  2.「CSVをアップロード」を選択します  3.「連絡先を追加」または「連絡先を追加して既存のリストに含める」オプションを選択します  4. CSVファイルが選択されていません  5.「連絡先を保存」ボタンをクリックします | システムに「CSVファイルを選択してください」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_04 | 登録済みユーザーが新しい連絡先を作成するためのリクエスト | 1. [連絡先を追加]をクリックします  2.「CSVをアップロード」を選択します  3.「連絡先を追加」または「連絡先を追加して既存のリストに含める」オプションを選択します  4. CSVファイルが選択されます  5.「連絡先を保存」ボタンをクリックします | 新しい連絡先が作成されます | N/A | パス | 25/07/2019 |  |

Table <Test Case> 新しい連絡先の作成

#### 4.2.7. 新しいテンプレートの作成:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 登録済みユーザーが新しいテンプレートを作成するためのリクエスト | 1. [新しいテンプレート]をクリックします  2.何もドラッグアンドドロップするコンポーネント  3.「テンプレートを保存」ボタンをクリックします | システムに「少なくとも1つのコンポーネントをドラッグアンドドロップする必要があります」と表示される” | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 登録済みユーザーが新しいテンプレートを作成するためのリクエスト | 1. [新しいテンプレート]をクリックします  2.コンポーネントをドラッグアンドドロップする  3.テンプレート名に何も記入しない  4.「テンプレートを保存」ボタンをクリックします | システムに「このテンプレートの名前を入力してください」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_03 | 登録済みユーザーが新しいテンプレートを作成するためのリクエスト | 1. [新しいテンプレート]をクリックします  2.コンポーネントをドラッグアンドドロップする  3.テンプレート名を入力します  4.「テンプレートを保存」ボタンをクリックします | 新しいテンプレートが作成されます | N/A | パス | 25/07/2019 |  |

Table <Test Case>新しいテンプレートの作成

#### 4.2.8. プロファイルの更新:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test case description** | **Test case procedure** | **Expected output** | **Inter-dependence test case** | **Result** | **Test date** | **Note** |
| EMP\_01 | 登録ユーザーのプロファイル更新リクエスト | 1. [ユーザープロフィール]をクリックします  2.何も編集しない  3.「プロファイルの更新」ボタンをクリックします | システムに「何も変更されていません」と表示される | N/A | パス | 25/07/2019 |  |
| EMP\_02 | 登録ユーザーのプロファイル更新リクエスト | 1. [ユーザープロフィール]をクリックします  2. 1つ以上のフィールドを編集します  3.「プロファイルの更新」ボタンをクリックします | システムに「ユーザープロファイルが更新されました」と表示される | N/A | パス | 25/07/2019 |  |

Table <Test Case>新しいテンプレートの作成

# Software User’s Manual

## Installation Guide

### Setting up environment at server

#### Hardware requirements

|  |  |  |
| --- | --- | --- |
| **Server** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (8 Mbps) | Cable, Wi-Fi (50 Mbps or more) |
| **Operating System** | CentOS 6 | CentOS 7 or Ubuntu 18 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **Computer Memory** | 2GB RAM | 8GB or more |
| **Storage space** | 3GB | 5GB or more |

Table - Hardware requirements for Server

#### Software requirement

|  |  |  |
| --- | --- | --- |
| **Name** | **Name/Version** | **Description** |
| **Environment** | Java EE 8.0, Node v10 | Specification for developing web application |
| **Dependency manager, build tool** | Yarn v1.15.2, Gradle 5.3.1, Maven 3.6.1 | Dependency managers, build tools for development |
| **Operating System** | Window 10 or Ubuntu 16 or MacOS | Operating system and platform for development |
| **Modeling tool** | StarUML, Draw.io | Used to design diagram |
| **IDE** | IntelliJ IDEA 2018.1.5, Visual Studio Code 1.27.2 | Programming tools |
| **DBEMP** | MySQL 8.0 | Used to create & manage the database for development |
| **Web browser** | Chrome 69 or above. | Testing browser |
| **Mobile OS** | iOS 10 or above | Testing mobile device operating system |
| **Deployment Server** | Docker Compose 1.24.0 or above | Deployment environment for production |

Table - Software requirements for Server

### Deployment at server side

#### Prepare deployment package

* **Development**

Download and install Java  
Link download: <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html>

* + Download and install IntelliJ IDEA (Includes Gradle and Maven)  
    Link download:  
    [https://www.jetbrains.com/idea/download/#section=windows](https://www.jetbrains.com/idea/download/)
  + Download and install MySQL 8.0  
    Link download:  
    <https://dev.mysql.com/downloads/installer/>
  + Download and install NodeJS v10 (Inludes NPM 6)  
    Link download:  
    <https://nodejs.org/en/download/>
  + Download and install Yarn  
    Link download:  
    <https://yarnpkg.com/lang/en/docs/install/#windows-stable>
* **Production**  
  Download and install Docker Compose:  
  Link download: <https://docs.docker.com/compose/install/>

#### Configure Server before deploy

* Development:
  + MySQL 8.0: create new database and new user and grant all privileges on the new database to the new user.
  + Web API application:
    - Copy capstone-EMP/client-api/src/main/tomee/deploy-conf/ tomee.xml to capstone-EMP/client-api/src/main/tomee/deploy/tomee.xml
    - Edit connection string
  + Web Admin application:
    - Edit connection string in file: capstone-EMP/webadmin/src/main/resources/application.properties

### Deploy web application on server

#### Development

* Database:
  + Restore database from file capstone-EMP/database/init.sql
* Web API application:
  + Open IntelliJ IDEA and open folder capstone-EMP/client-api as a project
  + At the top menu, click Run button
* Web admin application:
  + Open IntelliJ IDEA and open folder capstone-EMP/webadmin as a project
  + At the top menu, click Run button
* Web client:
  + Open VSCode and open folder capstone-EMP/client-web
  + Open intergrated terminal and run command: “yarn” to install dependency packages and “yarn dev” to start web client for development

#### Production

* For web application (includes Web API, Web Admin, Web Client, database):
  + Open terminal at project folder and run command: “npm start”
  + Install dependency packages with command: “npm install”

## User Guide

### Web application

#### Create Campaign







|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Set up schedule |
| 2 | Choose list |
| 3 | Fill in sender name |
| 4 | Fill in email address |
| 5 | Fill in subject |
| 6 | Click “Design Email” button |
| 7 | Choose template |
| 8 | Save campaign |

Table - Software requirements for Server

#### Create Appointment









|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Fill in appointment name |
| 2 | Set up schedule |
| 3 | Choose list |
| 4 | Fill in sender name |
| 5 | Fill in email address |
| 6 | Fill in subject |
| 7 | Click “Design Email” button |
| 8 | Choose template |
| 9 | Send appointment |

#### Create Embed Form



|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Fill in form name |
| 2 | Choose group |
| 3 | Add custom field |
| 4 | Save form |

#### Create Group



|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Fill in group name |
| 2 | Fill in description |
| 3 | Create new group |

#### Add Contact Manually



|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Choose group |
| 2 | Fill in email |
| 3 | Fill in first name |
| 4 | Fill in last name |
| 5 | More field to add contact |
| 6 | Delete that field |
| 7 | Save contact |

#### Add contact by upload file CSV



|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Choose group |
| 2 | Choose CSV file |
| 3 | Save contact |

#### Create New Template



|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Drag and drop contents |
| 2 | Drag and drop blocks |
| 3 | Drag and drop body |
| 4 | Save template |

# F1. ソフトウェアユーザーマニュアル

## インストールガイド

### サーバーでの環境のセットアップ

#### ハードウェア要件

|  |  |  |
| --- | --- | --- |
| **サーバ** | **最小要件** | **お勧め** |
| **インターネット接続** | ケーブル、Wi-Fi（8 Mbps） | ケーブル, Wi-Fi (50 Mbps以上) |
| **オペレーティング・システム** | CentOS 6 | CentOS 7またはUbuntu 18 |
| **コンピュータープロセッサー** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **コンピュータメモリ** | 2GB RAM | 8GB以上 |
| **収納スペース** | 3GB | 5GB以上 |

#### ソフトウェア要件

|  |  |  |
| --- | --- | --- |
| **Name** | **Name/Version** | **Description** |
| **環境** | Java EE 8.0, Node v10 | Webアプリケーション開発の仕様 |
| **依存関係マネージャー、ビルドツール** | Yarn v1.15.2, Gradle 5.3.1, Maven 3.6.1 | 依存関係マネージャー、開発用のビルドツール |
| **オペレーティング・システム** | Window 10 or Ubuntu 16 or MacOS | 開発用のオペレーティングシステムとプラットフォーム |
| **モデリングツール** | StarUML, Draw.io | 図の設計に使用 |
| **IDE** | IntelliJ IDEA 2018.1.5, Visual Studio Code 1.27.2 | プログラミングツール |
| **DBEMP** | MySQL 8.0 | 開発用データベースの作成と管理に使用 |
| **ウェブブラウザ** | Chrome 69 or above. | ブラウザーのテスト |
| **モバイルOS** | iOS 10 or above | モバイルデバイスのオペレーティングシステムのテスト |
| **展開サーバー** | Docker Compose 1.24.0 or above | 実動用のデプロイメント環境 |

### サーバー側での展開

#### 展開パッケージを準備する

* **開発**

Javaをダウンロードしてインストールする  
リンクのダウンロード：<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html>

* + IntelliJ IDEAをダウンロードしてインストールします（GradleとMavenを含む）  
    リンクのダウンロード：  
    [https://www.jetbrains.com/idea/download/#section=windows](https://www.jetbrains.com/idea/download/)
  + MySQL 8.0をダウンロードしてインストールします  
    リンクのダウンロード：  
    <https://dev.mysql.com/downloads/installer/>
  + NodeJS v10（NPM 6を含む）をダウンロードしてインストールします  
    リンクのダウンロード：  
    <https://nodejs.org/en/download/>
  + Yarnをダウンロードしてインストールする  
    リンクのダウンロード：  
    <https://yarnpkg.com/lang/en/docs/install/#windows-stable>
* **製造**  
  Docker Composeをダウンロードしてインストールします。  
  リンクのダウンロード：
* <https://docs.docker.com/compose/install/>

#### 展開する前にサーバーを構成する

* 開発：
  + MySQL 8.0：新しいデータベースと新しいユーザーを作成し、新しいデータベースに対するすべての権限を新しいユーザーに付与します。
  + Web APIアプリケーション：
    - capstone-EMP / client-api / src / main / tomee / deploy-conf / tomee.xmlをcapstone-EMP / client-api / src / main / tomee / deploy / tomee.xmlにコピーします
    - 接続文字列を編集
  + Web管理アプリケーション：
    - ファイルの接続文字列を編集します：capstone-EMP / webadmin / src / main / resources / application.properties

### サーバーにWebアプリケーションをデプロイする

#### 開発

* データベース：
  + ファイルcapstone-EMP / database / init.sqlからデータベースを復元します
* Web APIアプリケーション：
  + IntelliJ IDEAを開き、capstone-EMP / client-apiフォルダーをプロジェクトとして開きます
  + 上部のメニューで[実行]ボタンをクリックします
* Web管理アプリケーション：
  + IntelliJ IDEAを開き、capcap-EMP / webadminフォルダーをプロジェクトとして開きます
  + 上部のメニューで[実行]ボタンをクリックします
* Webクライアント:
  + VSCodeを開き、capstone-EMP / client-webフォルダーを開きます
  + 統合ターミナルを開き、コマンド「yarn」を実行して依存パッケージをインストールし、「yarn dev」を使用して開発用のWebクライアントを起動します

#### 製造

* Webアプリケーション（Web API、Web管理、Webクライアント、データベースを含む）の場合：
  + プロジェクトフォルダーでターミナルを開き、コマンド「npm start」を実行します
  + 「npm install」コマンドで依存関係パッケージをインストールします

## ユーザーガイド

### Webアプリケーション

#### キャンペーンを作成







|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | スケジュールを設定する |
| 2 | リストを選択 |
| 3 | 送信者名を入力してください |
| 4 | メールアドレスを入力してください |
| 5 | 件名を記入 |
| 6 | 「メールをデザイン」ボタンをクリックします |
| 7 | テンプレートを選択 |
| 8 | キャンペーンを保存 |

#### 予定を作成









|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | 予定名を入力してください |
| 2 | スケジュールを設定する |
| 3 | リストを選択 |
| 4 | 送信者名を入力してください |
| 5 | メールアドレスを入力してください |
| 6 | 件名を記入 |
| 7 | 「メールをデザイン」ボタンをクリックします |
| 8 | テンプレートを選択 |
| 9 | 予定を送信 |

#### 埋め込みフォームを作成



|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | 埋め込みフォームを作成 |
| 2 | グループを選択 |
| 3 | カスタムフィールドを追加 |
| 4 | フォームを保存 |

#### グループを作る



|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | グループ名を入力してください |
| 2 | 説明を入力してください |
| 3 | 新しいグループを作成 |

#### 連絡先を手動で追加



|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | グループを選択 |
| 2 | メールアドレスを入力してください |
| 3 | 名を記入してください |
| 4 | 姓を入力してください |
| 5 | 連絡先を追加するその他のフィールド |
| 6 | そのフィールドを削除する |
| 7 | 連絡先を保存 |

#### アップロードファイルCSVで連絡先を追加



|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | グループを選択 |
| 2 | CSVファイルを選択 |
| 3 | 連絡先を保存 |

#### 新しいテンプレートを作成



|  |  |
| --- | --- |
| **ステップ** | **説明** |
| 1 | コンテンツをドラッグアンドドロップする |
| 2 | ブロックをドラッグアンドドロップ |
| 3 | ボディをドラッグアンドドロップする |
| 4 | テンプレートを保存 |

# G. Appendix

1. **Spring Framework**

*[Online]* Available: https:/spring.io/docs/reference.

1. **MySQL**

*[Online]* Available: <https://dev.mysql.com/doc/>.

1. **ReactJS**

*[Online]* Available: <https://reactjs.org/docs/getting-started.html>

1. **React Native**

*[Online]* Available: <https://facebook.github.io/react-native/docs/getting-started>

1. **IntelliJ IDEA**

*[Online]* Available: <https://www.jetbrains.com/idea/documentation/>

1. **WebStorm**

*[Online]* Available: <https://www.jetbrains.com/webstorm/features/>

1. **Visual Studio Code**

*[Online]* Available: <https://code.visualstudio.com/docs>

1. **Nodejs**

*[Online]* Available: <https://nodejs.org/en/docs/>