|  |
| --- |
|  |

**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 –**

**

**CAPSTONE PROJECT REGISTER**

Class: Duration time: from …./…/…. To …/… /…..

(\*) Profession: <Software Engineer> Specialty: <ES> <IS> <JS>

X

(\*) Kinds of person make registers: Lecturer Students

X

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Lam Huu Khanh Phuong |  | phuonglhk@fe.edu.vn | Mr. |

2. Register information for students (if have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full name** | **Student code** | **Phone** | **E-mail** | **Role in Group** |
| Student 1 |  |  |  |  |  |
| Student 2 |  |  |  |  |  |
| Student 3 |  |  |  |  |  |
| Student 4 |  |  |  |  |  |

3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

* English: Email marketing platform with Simple Email Service of Amazon Cloud
* Vietnamese: EMP
* Abbreviation: EMP

(\*) 3.2. Main proposal content (including result and product)

1. Theory and practice (document):

* Student should apply the software development process and UML 2.0 in modelling system.
* The documents include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
* Server side technologies:
  + Server: .NET, Java or JavaScript, Windows Azure…
  + Database Design: SQL Server or MySQL.
* Client side technologies:
  + Web Client: HTML5, CSS3, JavaScript.

1. Program:

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.

- For sending a large amount of email from one sender to many recipients, you have to use the service of Email Service Provider, such as: Maichimp, Amazone SES, GetRespones.

- Amazon Simple Email Service (Amazon SES) is a cloud-based email sending service designed to help digital marketers and application developers send marketing, notification, and transactional emails. It is a reliable, cost-effective service for businesses of all sizes that use email to keep in contact with their customers.

- In this project, you have to build a web system which uses Amazon SES to send email marketing. Besides that, you must build some others features includes:

- Create and manage email content in HTML format.

-Dynamic content for each email with replaceable keywords (@address, @fullname, @birthday)

- Create and manage email marketing campaigns

- Meaning analysis and auto response email

- Segmentation email list.

- Optimize the time to send email to keep the high open-rate.

4. Other comment (propose all relative thing if have)

N/A

|  |  |
| --- | --- |
| **Supervisor (If have)**  *(Sign and full name)* | HCM, date …… ………. /20 …..  **On behalf of Registers**  *(Sign and full name)* |

**Table of Contents**

1. Contents

[**Table of Contents** 5](#_Toc17060384)

[List of table 7](#_Toc17060385)

[List of figure 7](#_Toc17060386)

[A. Introduction 8](#_Toc17060387)

[1. Project Information 8](#_Toc17060388)

[2. Introduction 8](#_Toc17060389)

[3. Current Situation 8](#_Toc17060390)

[4. Problem Definition 9](#_Toc17060391)

[5. Proposed Solution 9](#_Toc17060392)

[5.1. Functions 9](#_Toc17060393)

[5.2. Advantages and disadvantages 10](#_Toc17060394)

[6. Functional Requirements 10](#_Toc17060395)

[6.1. Email marketing website 10](#_Toc17060396)

[6.2. Admin website 11](#_Toc17060397)

[7. Role & Responsibility 11](#_Toc17060398)

[B. Software Project Management Plan 11](#_Toc17060399)

[1. Problem Definition 11](#_Toc17060400)

[1.1. Name of this Capstone Project 11](#_Toc17060401)

[1.2. Problem Abstract 12](#_Toc17060402)

[1.3. Project Overview 12](#_Toc17060403)

[2. Project Organization 15](#_Toc17060404)

[2.1. Software Process Model 15](#_Toc17060405)

[2.2. Roles and responsibilities 15](#_Toc17060406)

[2.3. Tools and Techniques 16](#_Toc17060407)

[3. Project Management Plan 17](#_Toc17060408)

[3.1. Product Backlog 17](#_Toc17060409)

[3.2. Sprint Backlog 17](#_Toc17060410)

[3.3. Deliverables 17](#_Toc17060411)

[3.4. All Meeting Minutes 17](#_Toc17060412)

[4. Coding Convention 17](#_Toc17060413)

[C. Software Requirement Specification 18](#_Toc17060414)

[1. User Requirement Specification 18](#_Toc17060415)

[1.1. Guest Requirement 18](#_Toc17060416)

[1.2. User requirement 18](#_Toc17060417)

[1.3. Admin requirement 19](#_Toc17060418)

[2. System Requirement Specification 20](#_Toc17060419)

[2.1. External Interface Requirement 20](#_Toc17060420)

[2.2. System Overview Use Case 20](#_Toc17060421)

[3. Conceptual Diagram 22](#_Toc17060422)

[D. Software Design Description 23](#_Toc17060423)

[1. Design Overview 23](#_Toc17060424)

[2. System Architecture Design 24](#_Toc17060425)

[3. Component Diagram 25](#_Toc17060426)

[4. Detailed Description 26](#_Toc17060427)

[4.1. Class Diagram 26](#_Toc17060428)

[4.2. Class diagram explaination 27](#_Toc17060429)

[4.3. Interaction 29](#_Toc17060430)

[5. Database design 32](#_Toc17060431)

[5.1. Entity realationship diagram 32](#_Toc17060432)

[5.2. Data dictionary 32](#_Toc17060433)

[6. Database Relationship Diagram 34](#_Toc17060434)

[6.1. Physical diagram 34](#_Toc17060435)

[6.2. Data Dictionary 35](#_Toc17060436)

[7. Algorithms 36](#_Toc17060437)

[7.1. Adjust the level of contact with the result returned of amazon service by using linear programming algorithm 36](#_Toc17060438)

# List of table

[Table 1: Roles and Responsibilities 11](#_Toc17063604)

[Table 2: Hardware Requirements for Server For PC 14](#_Toc17063605)

[Table 3: Hardware requirement for PC 14](#_Toc17063606)

[Table 4: Development Environment - Software Requirement 15](#_Toc17063607)

[Table 5: Roles and responsibilities 16](#_Toc17063608)

[Table 6: Tools and techniques 17](#_Toc17063609)

[Table 7: Deliverables 17](#_Toc17063610)

[Table 8: Component Data Dictionary 26](#_Toc17063611)

[Table 9 - <Class Diagram>Campaign Attributes 28](#_Toc17063612)

[Table 10- <Class Diagram> Campaign Methods 28](#_Toc17063613)

[Table 11- <Class Diagram>Appointment Attributes 29](#_Toc17063614)

[Table 12- <Class Diagram> Appointment Methods 29](#_Toc17063615)

[Table 13: <Class Diagram> Embedded Form 30](#_Toc17063616)

[Table 14:<Class Diagram> Workflow 30](#_Toc17063617)

[Table 15 Entity Relationship Diagram Data dictionary 36](#_Toc17063618)

[Table 16 Physical Diagram Dictionary 39](#_Toc17063619)

# List of figure

[Figure 1 - Scrum Framework 16](#_Toc17063633)

[Figure 2 - System overview use case 22](#_Toc17063634)

[Figure 3 - Conceptual Diagram 22](file:///E:\EmailMarketing\EmailMarketing\Documents\Mindsending%20report\%5bbrief%5d%20MindSending%20Report.docx#_Toc17063635)

[Figure 4 - System Architecture diagram 24](#_Toc17063636)

[Figure 5 - Component Diagram 25](#_Toc17063637)

[Figure 6 – Class Diagram 27](#_Toc17063638)

[Figure 7: Sequence Diagram - Login 31](#_Toc17063639)

[Figure 8: Sequence Diagram - Create Campaign 31](#_Toc17063640)

[Figure 9: Sequence Diagram - Send Campaign 32](#_Toc17063641)

[Figure 10: Sequence diagram – Create Workflow 33](#_Toc17063642)

[Figure 11: Sequence diagram – Pause Workflow 34](#_Toc17063643)

[Figure 12: Entity Relation Diagram 35](#_Toc17063644)

[Figure 13: Physical diagram 37](#_Toc17063645)

# 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 1: 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, 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 problem EMP 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

#### Future Plans

* Current system can only support user to view a few statistics. We design the system to provide more statistics analytic views for user:
  + User can view statistics and visualized graph about detail of each campaign, detail of all campaign, detail of each contact, detail of all contact
  + User can compare daily campaign with weekly and monthly campaign
* Current system can only suggest typed subject line to Registered User based on what user had used before. We design the system to support for further machine learning features:
  + Suggest subject line based on personalization, most uses subject line, trending subject line
* Current system can only provide 5 level to contact. We design the system to support more level to contact’s level:
  + Support more chance for user to improve contact score
  + Support more level and develop ranking system for contact

##### 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 2: 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 3: 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 4: 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 * Arrange Meeting * Managing project * Risk Management |
| 2 | Nguyễn Hữu Thắng | Scrum Master | * Designing databases * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| 3 | Nguyễn Lưu Hồng Sơn | Scrum team member | * Clarifying requirements * Preparing documents * Web GUI Design * Coding * Testing |

Table 5: 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 6: Tools and techniques

## Project Management Plan

### Product Backlog

Product Backlog could be found [here.](https://docs.google.com/spreadsheets/d/1frlWkSucBGWdu4LXIdjJmvkrjb7n8INJIswCkdlO8l0/edit#gid=1730897580)

### Sprint Backlog

Sprint Backlog can be found [here](https://docs.google.com/spreadsheets/d/1V7nzXLQfnIm1UGQAxpV4zYPuq_cqFh3VMB5n4_PapHo/edit#gid=242387435)

### 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 7: Deliverables

### All Meeting Minutes

All sprint meeting minutes could be found [here](https://drive.google.com/open?id=1W6nSgrdSwLY6ehYLZYo4inxbyWaHklfy).

## 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:*

***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

* Manage appointment:

-Add appointment

* Draw BPNM model:

-Add BPNM

-Pause BPNM

* Manage embed form

-Copy script

### 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 template

## 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.

### System Overview Use Case

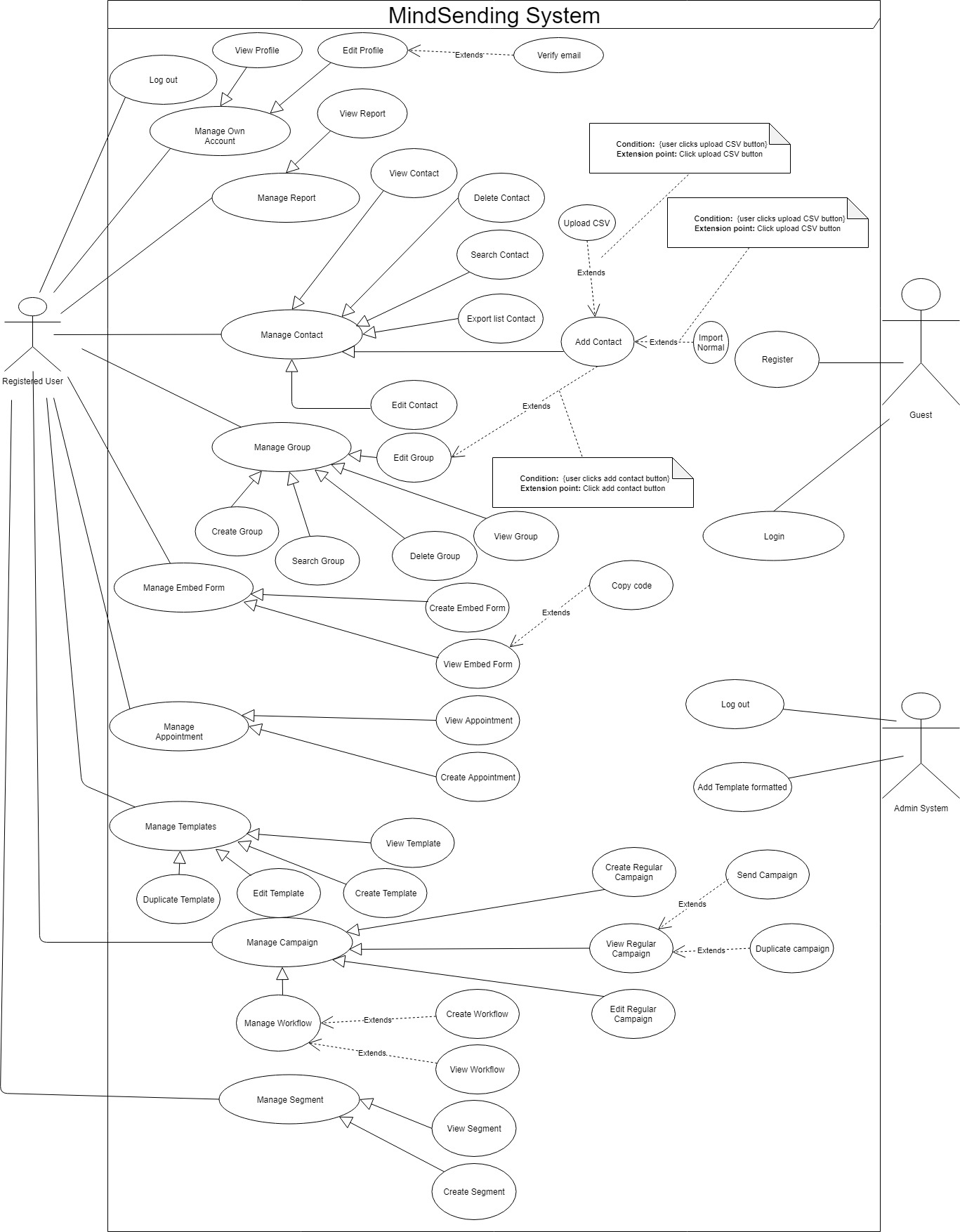
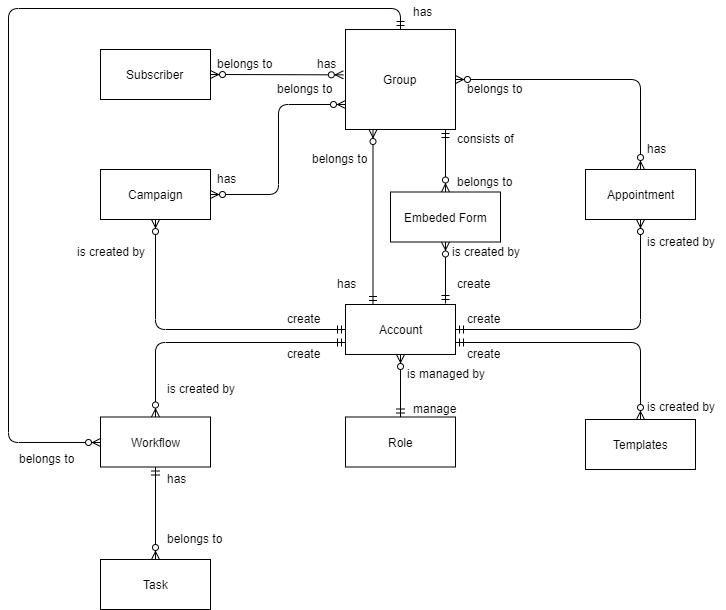


Figure 2 - System overview use case

## Conceptual Diagram

Figure 3 - Conceptual Diagram



# 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 4 - System Architecture diagram

## Component Diagram

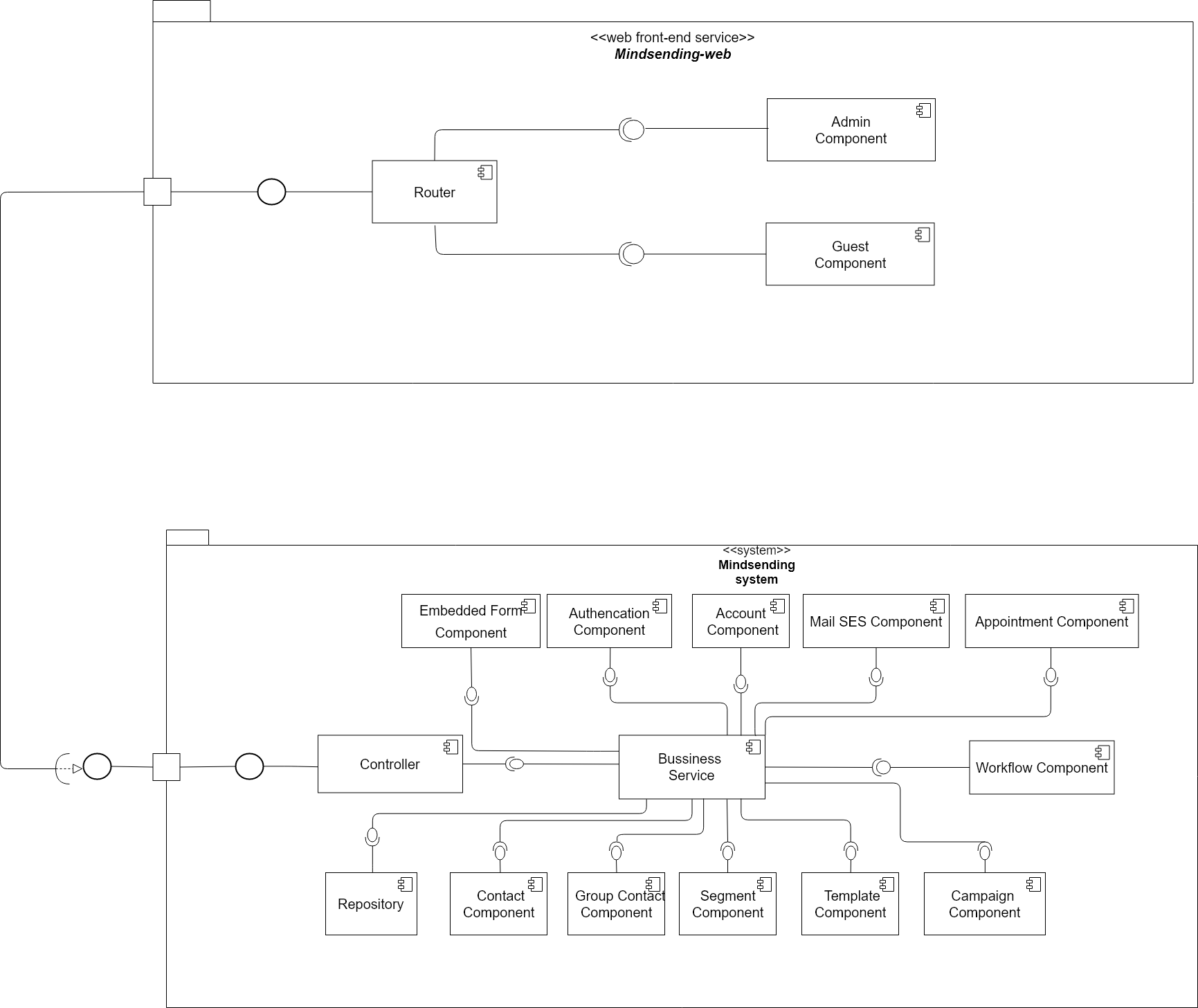


Figure 5 - Component Diagram

|  |  |
| --- | --- |
| **Component dictionary** | |
| **Component name** | **Description** |
| Embedded Form Component | A service responsible for managing information related to form |
| Appointment Component | A service responsible for managing information related to appointment |
| Mail SES Component | A service responsible for sending mail, receiving notifications and storage. |
| Account Component | A service responsible for managing information related to Account User |
| Contact Component | A service responsible for managing information related to contacts. |
| Authentication Component | A service responsible for authentication and authorization. |
| Workflow Component | A service responsible for run and create process of campaign |
| Group Contact Component | A service responsible for group contact |
| Segment Component | A service responsible for segment |
| Repository | A service responsible for data management. |
| Template Component | A service responsible for managing information related to templates |
| Admin Component | A component related to admin |
| Guest Component | A component related to Guest |

Table 8: Component Data Dictionary

## Detailed Description

### Class Diagram

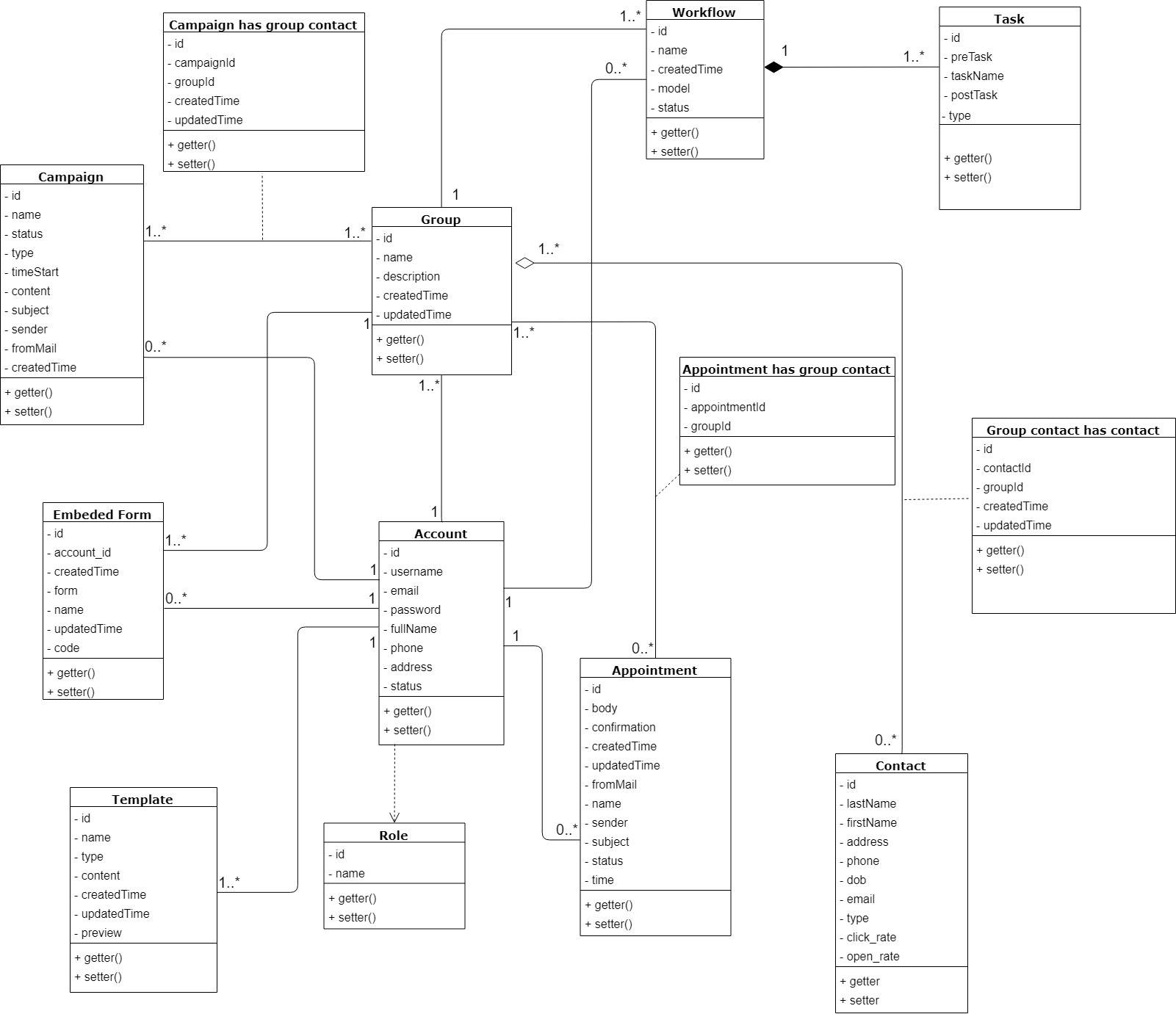


Figure 6 – Class Diagram

### Class diagram explaination

#### 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 9 - <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 10- <Class Diagram> Campaign 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 11- <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 12- <Class Diagram> Appointment 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 13: <Class Diagram> Embedded Form

#### 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 14:<Class Diagram> Workflow

### Interaction

#### Login



Figure 7: Sequence Diagram - Login

#### Create Campaign



Figure 8: Sequence Diagram - Create Campaign

#### Send Campaign

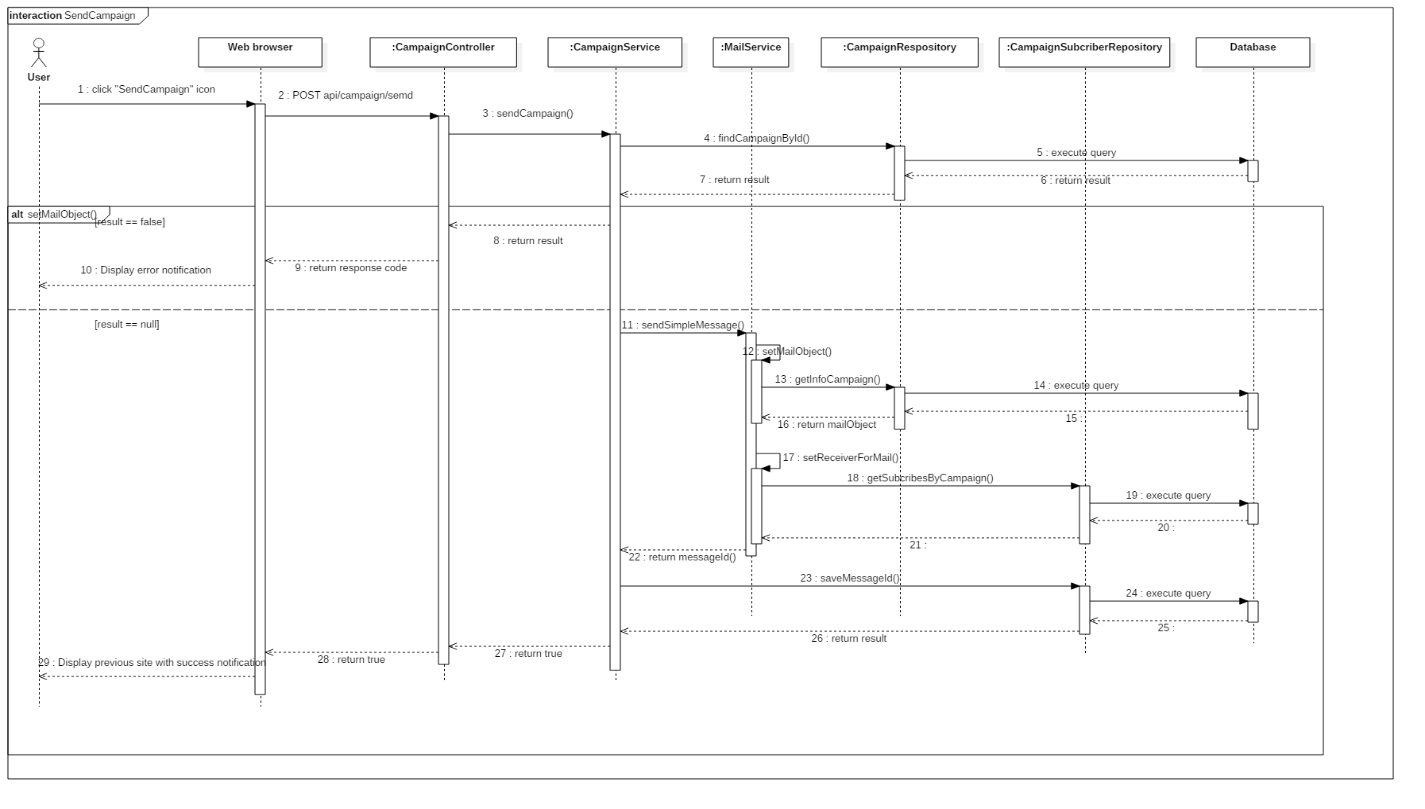


Figure 9: Sequence Diagram - Send Campaign

#### Create Workflow



Figure 10: Sequence diagram – Create Workflow

#### Pause Workflow



Figure 11: Sequence diagram – Pause Workflow

## Database design

### Entity realationship diagram

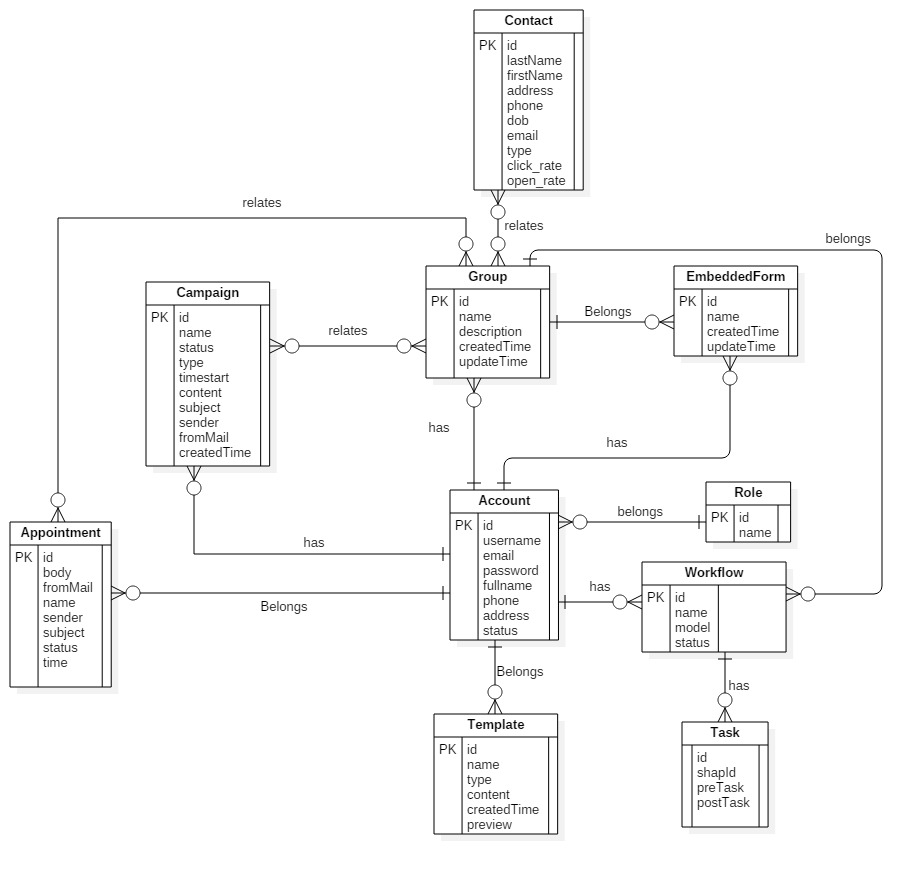


Figure 12: Entity Relation Diagram

### 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 15 Entity Relationship Diagram Data dictionary

## Database Relationship Diagram

### Physical diagram

Figure 13: 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 16 Physical Diagram Dictionary

## Algorithms

### 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

