Chocoholics Anonymous

Software Design Document

Version <1.2>

Zachary T. Brunette

Bryant M. Graff

Benjamin C. Haos

2016-12-11

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| 2016/10/12 | <1.0> | Initial Document | Zachary Brunette, Bryant Graff, Benjamin Haos |
| 2016/11/16 | <1.1> | Revision to new document standard | Bryant Graff |
| 2016/12/11 | <1.2> | Added UML diagrams; revised content | Bryant Graff |

Table of Contents

[Introduction](#_gjdgxs)

[Purpose](#_30j0zll)

[Scope](#_1fob9te)

[Definitions, Acronyms, and Abbreviations](#_3znysh7)

[Overview](#_2et92p0)

[Use Cases](#_tyjcwt)

[Actors](#_3dy6vkm)

[Member](#_1t3h5sf)

[Provider](#_4d34og8)

[Manager](#_2s8eyo1)

[List of Use Cases](#_17dp8vu)

[Manager – Get All Users](#_3rdcrjn)

[Manager – Get Provider Directory](#_26in1rg)

[Manager – Get Manager Report](#_lnxbz9)

[Manager – Get Provider Report](#_35nkun2)

[Manager – Get Member Report](#_1ksv4uv)

[Provider – Provide Service](#_44sinio)

[Provider – Login](#_2jxsxqh)

[Provider – Verify Member](#_z337ya)

[Provider – Charge for Service](#_3j2qqm3)

[Provider – Get Provider Report](#_1y810tw)

[Member – Get Member Report](#_2xcytpi)

[Use Case Diagrams](#_1ci93xb)

[Use Case Overview Diagram](#_3whwml4)

[Use Cases](#_2bn6wsx)

[Manager – Get All Users](#_qsh70q)

[Manager – Get Manager Report](#_3as4poj)

[Manager – Get Provider Directory](#_49x2ik5)

[Manager – Get Provider Report](#_2p2csry)

[Manager – Get Member Report](#_147n2zr)

[Provider – Provide Service](#_3o7alnk)

[Provider – Login](#_23ckvvd)

[Provider – Verify Member](#_ihv636)

[Provider – Charge for Service](#_32hioqz)

[Provider – Get Provider Report](#_1hmsyys)

[Provider – Get Provider Directory](#_41mghml)

[Member – Get Member Report](#_2grqrue)

[Design Overview](#_vx1227)

[Introduction](#_3fwokq0)

[System Architecture](#_1v1yuxt)

[System Interfaces](#_4f1mdlm)

[User Interface](#_2u6wntf)

[Data Interface](#_19c6y18)

[Constraints and Assumptions](#_3tbugp1)

[List of Assumptions](#_28h4qwu)

[List of Dependencies](#_nmf14n)

[Data Definitions](#_37m2jsg)

[Member](#_1mrcu09)

[Provider](#_46r0co2)

[Service](#_2lwamvv)

[Billable](#_111kx3o)

[System Object Model](#_3l18frh)

[Introduction](#_206ipza)

[Subsystems](#_4k668n3)

[Object Descriptions](#_2zbgiuw)

[Repository Classes](#_1egqt2p)

[Entity Classes](#_3ygebqi)

[Application and Controller Classes](#_2dlolyb)

[Design Diagrams](#_sqyw64)

[Create Billable – Sequence Diagram](#_3cqmetx)

[Supplementary Documentation](#_1rvwp1q)

[Tools Used to Create Diagrams](#_4bvk7pj)

Software Design Document

# Introduction

Chocoholics Anonymous has requested an application to track the cost and usage of its services amongst its members and providers. This document will define the problem and detail the steps necessary to create such an application.

## Purpose

The Chocoholics Anonymous Software (CAS) is designed to handle data processing for Chocoholics Anonymous. The CAS must record member information as well as information on services rendered. It must also keep information on the services provided by Chocoholics Anonymous as well as the providers of these services. Additionally, this information is to be disseminated via reports which are to be distributed to members and providers. A report is also to be generated weekly to be viewed by the appropriate managing authority.

## Scope

This scope of this program is limited to data processing. The following features are explicitly outside of the scope of this document:

* Communications software
* ChocAn provider terminal
* Acme Accounting software
* Implementation of the EFT component

## Definitions, Acronyms, and Abbreviations

**CAS** – Chocoholics Anonymous Software

**EFT**- Electronic Funds Transfer

## Overview

This document is divided into 8 sections. The sections of this document are as follows:

1. Introduction
2. Use Cases
3. Design Overview
4. Data Definitions
5. System Object Model
6. Object Descriptions
7. Design Diagrams
8. Supplementary Documentation

# Use Cases

## Actors

### Member

The member is the individual that purchases services from Chocoholics Anonymous through its providers. The member does not act directly on the system, but may receive a report on service usage. A member can be any individual with a valid account.

### Provider

The provider uses the provider console to charge members. The provider also receives a report detailing any services provided over a specified time period.

### Manager

The manager is someone that administers the Chocoholics Anonymous data center. The manager can create, delete and modify members, providers, and services. Additionally, the manager receives a weekly report on the usage of the system by both members and providers.

## List of Use Cases

### Manager – Get All Users

### Manager – Get Provider Directory

### Manager – Get Manager Report

### Manager – Get Provider Report

### Manager – Get Member Report

### Provider – Provide Service

### Provider – Login

### Provider – Verify Member

### Provider – Charge for Service

### Provider – Get Provider Report

### Provider – Get Provider Directory

### Member – Get Member Report

## Use Case Diagrams

### Use Case Overview Diagram



## Use Cases

### Manager – Get All Users

|  |  |
| --- | --- |
| Use Case | Get All Users |
| Actor | Manager |
| Description | A Manager must be able to access all Member information. |
| Precondition | The Manager must be a valid user. There must be at least one valid member in the system. |
| Workflow | 1. Select the “User” menu option 2. Select “Get All Users” 3. Shows all members and related information on the display. |
| Post condition | None |
| Acceptance Criteria | The Manager knows any pertinent information for all Members in the system. |

### Manager – Get Manager Report

|  |  |
| --- | --- |
| Use Case | Get Manager Report |
| Actor | Manager |
| Description | A Manager must have access for a weekly report on Chocoholics Anonymous services provided by provider. |
| Precondition | The Manager must be a valid user. |
| Workflow | * + - 1. Select the “Reports” menu option.       2. Select “Manager Report”       3. Select the desired week, denoted by the start date, from the drop-down box.       4. Click “Submit”       5. Shows a tabulated listing of all providers that provided services for the selected week as well as the number of consultations and the total fee from those consultations. |
| Alternate Workflow | 1. If no services have been provided, the drop-down box will be empty. |
| Post condition | None |
| Acceptance Criteria | The Manager has a weekly report that summarizes services provided for the selected week. |

### Manager – Get Provider Directory

|  |  |
| --- | --- |
| Use Case | Get Services Directory |
| Actor | Manager |
| Description | A Manager must be able to get a list of Services for Providers that are used to identify and describe Services. |
| Precondition | The Provider must be valid. |
| Workflow | 1. Select the “Service” menu option. 2. Select “Get All Services” 3. A list of all services will be displayed detailing each service that can be provided. |
| Alternate Workflow | None |
| Post condition | None |
| Acceptance Criteria | The Manager has a current list of available Services to be given to Providers. |

### Manager – Get Provider Report

|  |  |
| --- | --- |
| Use Case | Get Services Provided Report |
| Actor | Manager |
| Description | A Provider must be provided with a report detailing the services rendered within a specified time span, usually a week. |
| Precondition | The Provider must be valid. |
| Workflow | 1. Select the “Reports” menu option. 2. Select “Provider Report”. 3. Select the desired provider from the drop-down box. 4. Click “Submit” 5. Select the desired week, denoted by the start date, from the drop-down box. 6. Click “Submit” 7. Shows a listing of all services provided for the selected week as well as which members were services, when they were services and the fee charged for the service. |
| Alternate Workflow | **Alternate Workflow**   1. If there are no providers, the drop-down box will be empty   **Alternate Workflow**   1. If the provider has yet to provide services, the drop-down box will be empty. |
| Post condition | None |
| Acceptance Criteria | The Manager has a list of services provided by the selected Provider for the requested time span to send to the Provider. |

### Manager – Get Member Report

|  |  |
| --- | --- |
| Use Case | Get Member Report |
| Actor | Manager |
| Description | A Member must be provided with a report detailing the services used within a specified time span, usually a week. |
| Precondition | The Member must be valid. |
| Workflow | 1. Select the “Reports” menu option. 2. Select “User Report”. 3. Select the desired member from the drop-down box. 4. Click “Submit” 5. Select the desired week, denoted by the start date, from the drop-down box. 6. Click “Submit” 7. Shows a listing of all services provided for the selected week detailing when the service was provided, which Provider provided the service, and the description of that service. |
| Alternate Workflow | **Alternate Workflow**   1. If there are no users, the drop-down box will be empty   **Alternate Workflow**   1. If the provider has yet to provide services, the drop-down box will be empty. |
| Post condition | None |
| Acceptance Criteria | The Manager has a list of services provided to the selected Member for the requested time span to send to that Member. |

### Provider – Provide Service

|  |  |
| --- | --- |
| Use Case | Provide Service |
| Actor | Provider |
| Description | A Provider must be able to provide services for valid members. |
| Precondition | There must be at least one Provider and one Member in the system. Provider is currently logged in. |
| Workflow | 1. Begin provider communication terminal prompt shown. 2. User enters nine-digit member id. 3. Display member status associated with id as valid. 4. User signals to enter service provided record. 5. Prompt for date service provided. 6. Enter date MM-DD-YYYY. 7. Prompt for service code. 8. Enter six-digit service code. 9. Display service associated with code. 10. Prompt for optional comment on service. 11. Enter comment. 12. Prompt for verification. 13. Enter positive verification. 14. Write data to file. |
| Alternate Workflow | **Alternate Workflow**  2. Entry does not contain nine characters.  3. Display invalid input message sent to provider terminal.  4. Return to receive provider communication prompt. (line 1)  **Alternate Workflow**  2. Entry does not contain nine numbers.  3. Display invalid input message sent to provider terminal.  4. Return to receive provider communication prompt. (line 1)  **Alternate Workflow**  6, 8, 11, 13. (Correct format and size of input is verified in similar fashion for all data receiving lines.) |
| Post condition | The results of the transaction must be recorded. |
| Acceptance Criteria | Valid interactions are recorded in the system and contain the necessary information used to identify the Provider, Member and the service provided. |

### Provider – Login

|  |  |
| --- | --- |
| Use Case | Login |
| Actor | Provider |
| Description | A Provider must be able to log in to the system. |
| Precondition | There must be at least one Provider in the system. |
| Workflow | 1. The Provider enters his/her provider number 2. The provider number is verified |
| Alternate Workflow | 2. The provider number is invalid.  3. The provider console prompts Provider to re-enter provider number  4. Return to step 1. |
| Post condition | None. |
| Acceptance Criteria | The Provider is now logged into the system. |

### Provider – Verify Member

|  |  |
| --- | --- |
| Use Case | Verify Member |
| Actor | Provider |
| Description | A Provider must be able to verify if a Member is eligible to receive services. |
| Precondition | There must be at least one Member in the System. The Provider must be logged in. |
| Workflow | 1. The Provider scans member card using provider console. 2. The system verifies member account, displaying “Validated” if the account is valid. |
| Alternate Workflow | **Alternate Workflow**   1. Provider enters member number into provider console.   **Alternate Workflow**   1. System determines that member card is invalid.   **Alternate Workflow**   1. System determines the member account is suspended. |
| Post condition | None |
| Acceptance Criteria | The Member is now active in the provider console. |

### Provider – Charge for Service

|  |  |
| --- | --- |
| Use Case | Charge for Service |
| Actor | Provider |
| Description | A Provider must be able to charge for a service provided. |
| Precondition | There must be at least one service in the system. The Provider must be logged in. A Member account must be active in the provider console. The date and service code must be valid. |
| Workflow | 1. Provider enters date into the console in the following format: MM-DD-YYYY. 2. System verifies date. 3. Provider enters service code corresponding to the service provided 4. Provider console displays name of service provided and prompts Provider for confirmation. 5. Provider confirms information by selecting “Yes” 6. System records changes to the Member account |
| Alternate Workflow | **Alternate Workflow**   1. Provider enters an invalid date. Retry step 1.   **Alternate Workflow**   1. Console displays “Invalid Service” Retry step 3.   **Alternate Workflow**   1. Provider selects “No” Retry step 3. |
| Post condition | None |
| Acceptance Criteria | The transaction is recorded in the system and accurately reflects the service provided as well as the Provider, Member and Service information. |

### Provider – Get Provider Report

|  |  |
| --- | --- |
| Use Case | Get Report |
| Actor | Provider, Manager |
| Description | A Provider must be provided with a report detailing the services rendered within a specified time span, usually a week. |
| Precondition | The Provider must be valid. There must be at least one Provider in the system. |
| Workflow | 1. The Provider contacts a Manager to get the report. 2. The Manager obtains the report – see Manager: Get Provider Report 3. The Manager sends the Provider the report. |
| Post condition | None. |
| Acceptance Criteria | The Provider now has a report detailing services rendered for the specified time period. |

### Provider – Get Provider Directory

|  |  |
| --- | --- |
| Use Case | Get Report |
| Actor | Provider, Manager |
| Description | A Provider must be provided with a list of services that can be provided. |
| Precondition | The Provider must be valid. There must be at least one Provider in the system. |
| Workflow | 1. The Provider contacts a Manager to get the report. 2. The Manager obtains the report – see Manager: Get Provider Directory 3. The Manager sends the Provider the report. |
| Post condition | None. |
| Acceptance Criteria | The Provider now has a report detailing services rendered for the specified time period. |

### Member – Get Member Report

|  |  |
| --- | --- |
| Use Case | Get Member Report |
| Actor | Member, Manager |
| Description | A Member must be provided with a report detailing the services used within a specified time span, usually a week. |
| Precondition | The Member must be valid. There must be at least one Member in the system. |
| Workflow | 1. The Member contacts a Manager to get the report. 2. The Manager obtains the report – see Manager: Get Member Report 3. The Manager sends the Member the report. |
| Post condition | None. |
| Acceptance Criteria | The Member now has a report detailing services rendered for the specified time period. |

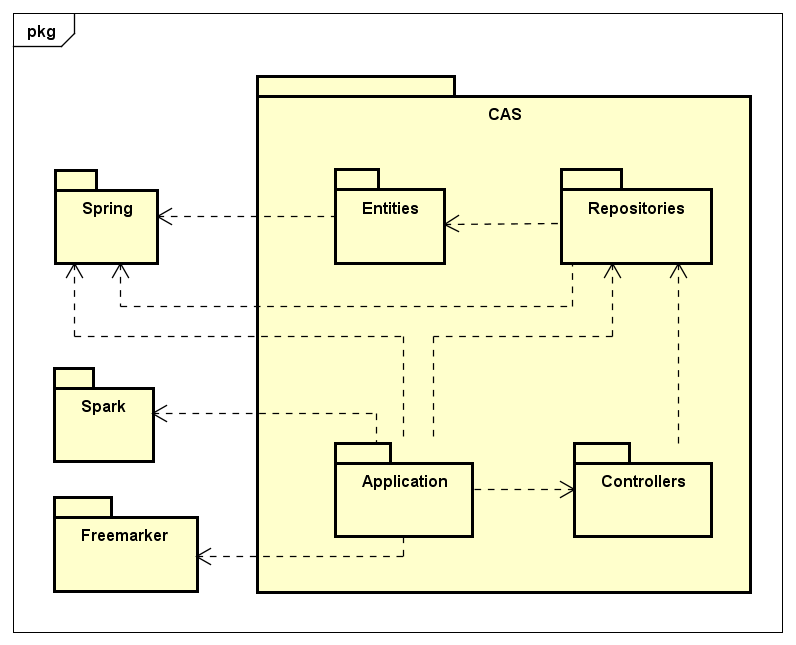
# Design Overview

## Introduction

The design of this system uses a model-view controller pattern. This allows for the user interface to be modified easily in anticipation of continued development without requiring any change to the underlying program logic or data. This design also incorporates several frameworks to aid in rapid development, which also translates to faster implementation times.

## System Architecture

This diagram provides a simplified view of the interactions between specific groupings of classes within the program and the various frameworks.



## System Interfaces

### User Interface

The user interface is provided through web content generated using the Freemarker, Spark, and Spring frameworks. This interface is designed for use in the data center, so the data is available through a simple series of menu choices. Entities can be added, modified, or deleted freely using this interface.

### Data Interface

The information handled by the software uses MongoDB through the Spring framework.

## Constraints and Assumptions

### List of Assumptions

It is assumed that Acme Accounting Services will provide file(s) needed to verify an existing user and last payment received date.

It is assumed that Acme Accounting Services will manage accepting payments from new members as well as current members.

It is assumed that communication time-outs will be handled by the contractor tasked with creating the communication software.

It is assumed that the data processing portion of the software will only record provider fee to be paid and not be in any way responsible for producing data relevant to money owed to provider.

It is assumed that information security will be handled by a third party.

### List of Dependencies

This system is dependent on the following items:

* The communication software the handles interactions between system and the provider console, as well as authorization and information security.
* The Acme accounting software interface’s ability to authenticate and communicate with the system.
* The integrity of the information communicated with the provider console.

# Data Definitions

## Member

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Description |
| memberName | String | 25 | The name of the member |
| entityUserIdNumber | Integer | 9 | Identification number of the member |
| entityUserEmailAddress | String | 50 | The email address of the member |
| memberStreetAddress | String | 25 | The member’s street address of residence |
| memberCity | String | 14 | The member’s city of residence |
| memberState | String | 2 | The member’s state of residence |
| memberZip | Long | - | The member’s zip code of residence |
| memberValidThrough | Date/Time | - | The duration left on membership |

## Provider

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Description |
| providerName | String | 25 | The name of the provider |
| entityProviderIdNumber | Integer | 9 | Identification number of the provider |
| entityProviderEmailAddress | String | 50 | The provider’s email address |
| providerStreetAddress | String | 25 | The provider’s street address of residence |
| providerCity | String | 14 | The provider’s city of residence |
| providerState | String | 2 | The provider’s state of residence |
| providerZip | Long | - | The provider’s zip code of residence |
| isDietitian | Boolean | - | True if the provider is a dietitian |
| IsExerciseExpert | Boolean | - | True if the provider is an exercise expert |
| IsInternist | Boolean | - | True if the provider is an internist |

## Service

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Description |
| entityServiceIdNumber | String | 6 | Identification number for the service |
| providableServiceDescription | String | 25 | A brief description of the service |
| isProvidableByDietitian | Boolean | - | True if the service is provided by a dietician |
| isProvidableByExerciseExpert | Boolean | - | True if the service is provided by an exercise expert |
| isProvidableByInternist | Boolean | - | True if the service is provided by an internist |

## Billable

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Description |
| entityBillableIdNumber | String | 12 | Identification number for the billable |
| dateServicedRecorded | Date/Time | - | Transaction recorded on this date and time |
| dateServiced | Date/Time | - | Actual date and time of transaction |
| providerNumberServicing | Integer | 9 | The ID of the provider servicing |
| memberNumberService | Integer | 9 | The ID of the member serviced |
| serviceNumberServiced | Integer | 6 | The ID of the service provided |
| serviceComment | String | 100 | Miscellaneous information |
| serviceCost | Double | - | The cost of the service provided |

# System Object Model

## Introduction

This software utilizes the following frameworks:

* Spring
* Spark
* Freemarker

Additionally, it uses Joda-Time for storing date and time information, as well as operations performed on such data. Jackson Data Processor, a general data-binding package, is used to parse JSON into entities.

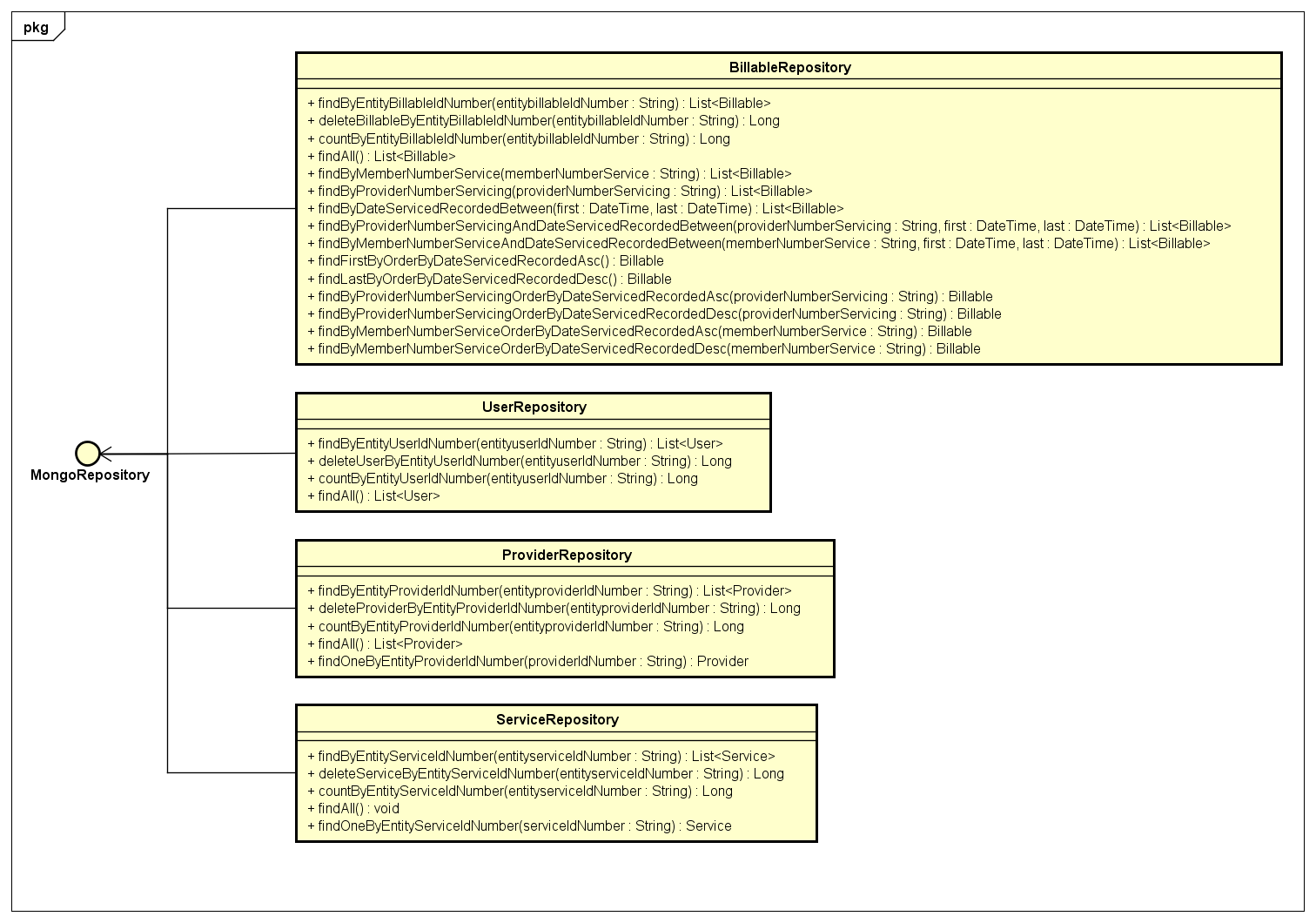
## Subsystems

All of the repository classes are generated through the use of Spring interfaces. The code in these repository classes describe queries that are used to select records from the database. The database uses MongoDB through the Spring framework.

The Freemarker engine renders HTML from Route output that is defined in the Application class.

# Object Descriptions

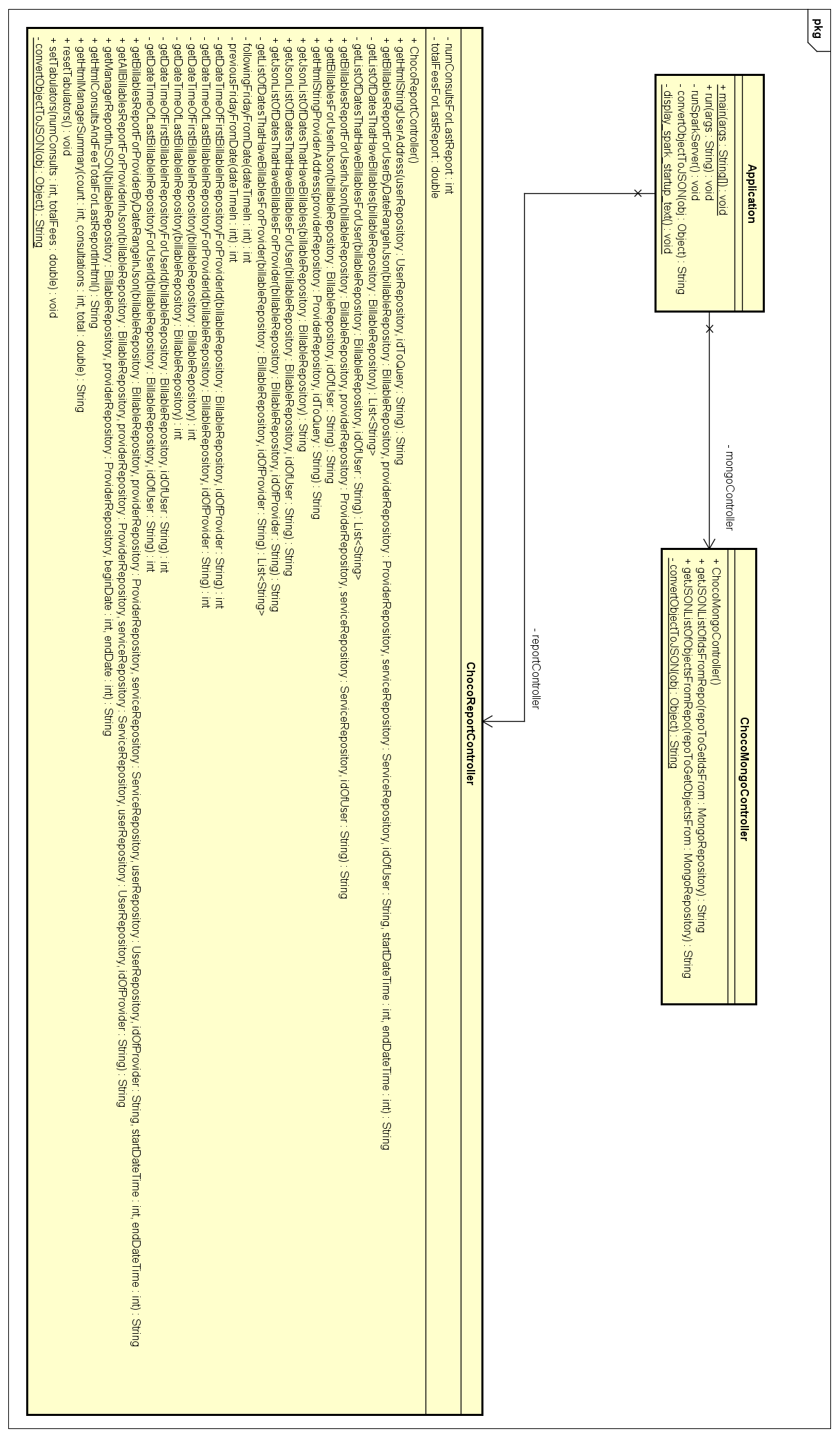
### Repository Classes



### Entity Classes

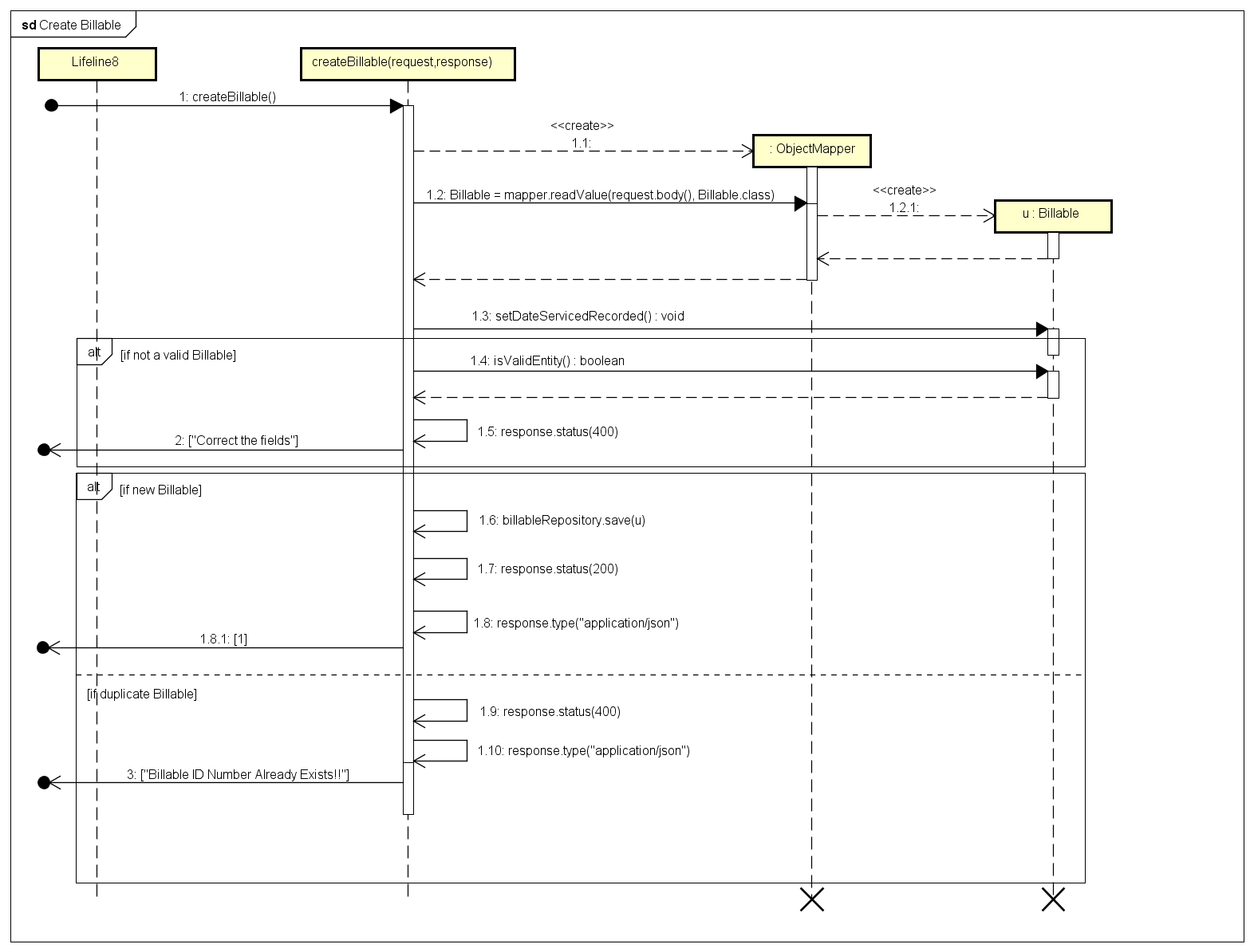


### Application and Controller Classes



# Design Diagrams

## Create Billable – Sequence Diagram



# Supplementary Documentation

## Tools Used to Create Diagrams

Astah Professional - <http://astah.net/editions/professional>

Draw.io - <https://www.draw.io/>