

# ソフトウェア設計演習 最終レポート

222C1021 今村優希

2024 年 8 月 16 日

## 1 今回のシステム

今回は下記仕様の「Live Campus」のようなシステムをモデル化する．システムの大まかな概要を下記に記す．

- 学生は毎学期ごとに、システムから履修登録を行い、許可された講義を受講する。履修の可否は、学修細則に基づきシステムが判断する。各学期ごとに履修できる科目数には上限があり、同じ時限に重複した科目を履修することはできない。
- 学生は、履修登録期間内には何度でも登録、削除、修正 (更新) を行うことが出来る。
- 学生は、登録した科目と登録可能な科目の時間割をそれぞれ見ることが出来る。
- 学生は、自身のこれまでの成績を確認することができる。
- 科目には、必修と選択があり、不合格の必修科目があれば履修登録時にシステムが提示し、履修を促す。
- 教員は、複数の科目を担当することがあり、システムから各科目の成績を報告する。なお、科目担当者は 1 名とする。
- システムは、成績報告期限までに報告していない教員に督促メールを送る。なお、成績報告期限の 1 週間前に成績報告していない教員には、期限を知らせるメールを送るものとする。

## 2 設計の作成手順

今回の設計では、分析モデルを作成した後に、設計モデルを作成する手順を取った。

分析モデルでは以下の UML 図を作成した。各 UML 図が作成し終わると、レビューを行い、後の工程に大きな影響が出ないように工夫を行った。

- ユースケース図
- ユースケース記述
- クラス図
- シーケンス図
- アクティビティ図
- ステートマシン図

### 3 ユースケース図

システムの概要を把握するためにユースケース図を作成した。

#### 作成

今回のシステムの内容から、必要なアクターは

- 学生
- 教員

であると考えた。学生は

#### レビュー結果メモ

- ユースケースが多く詳細過ぎると思ったので、一部ユースケースを統合した。
- アクターとして「外部システム」という名称でアクターを追加していたが、何の役割があるのかははっきりしていなかったため、「チケット購入補助」に名称を変更した。
- アクターとして「サーバー」を追加していたが、システム内に組み込まれていると把握したので削除した。

#### 作成結果

これらの考えのもと作成したユースケース記述が図である。付け加えた機能のために考えたユースケースを青で表示した。

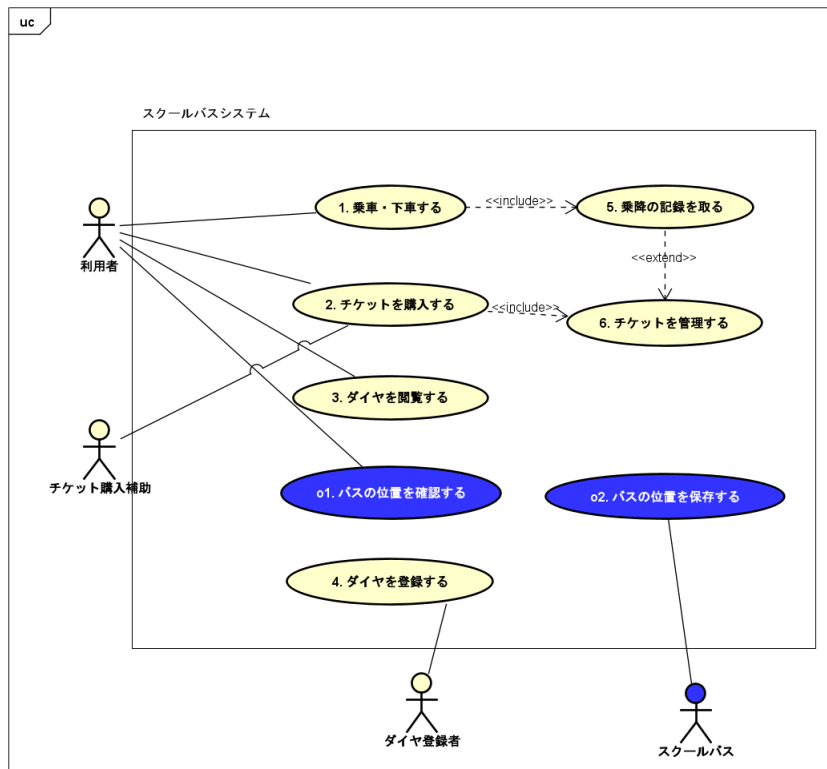


図 1 ユースケース図

## 4 ユースケース記述

上記で作成したユースケースすべてに対してユースケース記述の作成を行った。

### 概要と作成結果

#### 1. 乗車・下車

乗車し，降車するまでの一連の流れを記述で確認した．作成したユースケース記述は図である．

項目	内容
ユースケース	1. 乗車・下車する
基本系列	1-1. 利用者はスマホにチケットを表示する 1-2. 利用者はスマホをかざす 1-3. 読み取り機がサーバーに問い合わせる 1-4. サーバーが許可を出す(E1-1) 1-5. 利用者が乗車する 1-6. 利用者がスマホにチケットを表示する 1-7. 利用者はチケットをかざす 1-8. 読み取り機はサーバーに通知する(E1-2) 1-9. 乗客は下車する
代替系列	E1-1. 許可がなかった場合ブザーを出す E1-2. 通知失敗すると一定時間待って再度通知する

図 2 乗車・下車のユースケース記述

#### 2. チケットを購入する

チケットを購入するまでの流れをユースケース記述で確認した．利用者はサーバーを通じてチケットの購入結果がわかるのでシステムのバグが減るよう工夫した．

項目	内容
ユースケース	2. チケットを購入する
基本系列	2-1. 利用者はチケットを選ぶ(E2-1) 2-2. 外部システム(生協アプリ，クレジットカード)に移動する 2-3. 利用者は外部システムで購入手続きをする(E2-2) 2-4. 外部システムが購入を確認しサーバーに通知する(E2-3) 2-5. サーバーが利用者に購入通知を出す
代替系列	E2-1. 適切な選択ができなかったらエラーを通知する E2-2. 外部システムで購入が失敗すると失敗通知とともに初期に戻る E2-3. 通知に失敗すると，一定時間をおいて再度通知を試みる

図 3 チケット購入のユースケース記述

#### 3. ダイヤを確認する

ダイヤ確認のユースケース記述を作成した．

項目	内容
ユースケース	3. ダイヤを閲覧する
基本系列	3-1. 利用者はwebアプリを用いてダイヤ確認通知を出す 3-2. サーバーはダイヤをアップロードする(E3-1) 3-3. 利用者はwebアプリを用いてダイヤを確認する(E3-2)
代替系列	E3-1. アップロードを失敗すると失敗通知をwebアプリに送る E3-2. 表示を失敗すると失敗と表示する

図 4 ダイヤ確認のユースケース記述

#### 4. ダイヤを登録する

大学の事務が行うダイヤ登録をユースケース記述で動作を確認した。

項目	内容
ユースケース	4. ダイヤを登録する
基本系列	4-1. 大学の事務はダイヤを作成する 4-2. 大学の事務はダイヤをサーバーにアップロードする(E4-1) 4-3. サーバーはダイヤを保存する
代替系列	E4-1. アップロード失敗すると失敗通知を出す

図 5 ダイヤ登録のユースケース記述

#### 5. 乗降の記録と取る

1 によって利用者が乗車・下車を行ったのでその情報を保存するための記述を作成した。このユースケースの動作は、システムに内包されている「読み取り機」からの通知を元にサーバー側での動作がメインである。したがって、失敗がない前提で考えた。下車することでチケットを更新することで、システム場でのエラーを回避するよう工夫した。また、チケットの更新は 6. チケットの管理と協力して行うように工夫した。

項目	内容
ユースケース	5. 乗降の記録を取る
基本系列	5-1. 読み取り機からサーバーに乗車問い合わせが来る 5-2. 問い合わせ結果を読み取り機に返す 5-3. 乗車情報を保存する 5-4. 読み取り機からサーバーに降車通知がある 5-5. 降車情報を保存する 5-6. サーバーはチケットの更新を行う
代替系列	

図 6 乗降の記録を取るのユースケース記述

## 6. チケットを管理する

2によって利用者はチケットを購入し、5によってチケットが更新される。これらをまとめて管理するユースケース記述を作成した。

項目	内容
ユースケース	6. チケットを管理する
基本系列	6-1. 購入通知が来ると情報をサーバーに保存する(E6-1) 6-2. チケットの更新があると残りのチケット回数を減らす 6-3. チケットの回数が0になるとそのチケットをアーカイブにする
代替系列	E6-1. 保存ができないときは購入キャンセル通知を出す

図7 チケットを購入するのユースケース記述

### o1. バスの位置を確認する

利用者がバスの位置を得るまでの流れをユースケース記述で確認した。今回のシステムで記述した通り、位置情報は緯度経度で表すことにしたので、システムはかなりシンプルにできた。

項目	内容
ユースケース	o1. バスの位置を確認する
基本系列	o1-1. 利用者はwebアプリを利用してバス確認通知を出す o1-2. サーバーがバスの位置を確認する(Eo1-1) o1-3. サーバーが利用者にバスの位置をアップロードする
代替系列	Eo1-1. もしバスの位置が適当でなかった(確認できるバスがなかった)場合、エラー通知を出す

図8 バスの位置を確認するのユースケース記述

### o2. バスの位置を保存する

バス側で位置を取得すると、その情報をサーバーにアップロードする。位置取得は分単位と高頻度で行うとする。ここで工夫したのは、エラーが起きてもその対策とする代替系列を作成しなかったことだ。基本的に高頻度で情報がアップロードされるので、次の情報を待つようにした。

項目	内容
ユースケース	o2. バスの位置を保存する
基本系列	o2-1. バスが位置を取得する(Eo2-1) o2-2. バスが位置をサーバーに通知する(Eo2-2) o2-3. サーバーはバスの位置を更新する(Eo2-3)
代替系列	Eo2-1. 位置の取得ができなかったら再度位置取得を試みる Eo2-2. 通知を失敗するとその位置を通知することを諦める

図9 バスの位置を保存するのユースケース記述

## レビュー

- いくつかのユースケース記述 (2, 3) で、何を用いて動作を行っているかが欠如していた。今回は web アプリを用いて行うことにし、ユースケース記述の更新を行った。
- 代替系列ができていない部分があったので訂正を加えた。



## 5 クラス図

### 概要

システム概要とユースケース記述からクラス図の作成を行った。まず、その概要から挙げられたクラス図を以下に列挙する。ネストで表しているのは汎化関係 (g), 集約関係 (a) を表している。ネストで表しても大丈夫なのかわからなかったが、文章でわかりやすくするために今回はネストを使用した。

- システム利用者 このシステムを利用する人をまとめたクラスである。このクラスには乗客と事務と 2 つのクラスが関連している。このクラスには利用者 ID と利用者用の名前等のデータが保存されている。
  - a 乗客 乗客には乗客の名前がデータとして保存されている。メソッドとして、乗車・下車、ダイヤ確認、スマホをかざす等、乗客が直接行う動作を加えている。これらメソッドを利用するためのクラスが関連付けられている。
  - a 事務 事務は、ダイヤを登録する人を指すクラスである。ダイヤと関連付けられている。
- ダイヤ 事務と 1 乗客に関連付けられている。
- スマホ 乗客とチケットに関連付けられている。
- バス 乗客と関連付けられている。
  - g チケット読み取り機 チケットとサーバーと関連付けられている。チケットを読み取るとサーバーに通知が行き、サーバーが読み取り機に対して返答を出す。
- チケット スマホと関連付けられている
  - a 回数券
  - a 月利用券
- 外部システム チケットを購入するために使用する。サーバー関連づいている。
  - a 生協アプリ
  - a クレジット会社
- サーバー ダイヤ等を保存している。情報受付と行い、必要な情報をアップロードする。

それから、このシステムを動作させる上で必要だと思ったものをクラスとして定義を行った。

バス位置 バスの位置等を持つクラスである。サーバーや利用者が使うためクラスとして独立させた。

### レビュー結果メモ

- 複数のクラスが不要だったので削除した。同時に必要なクラスを追加した
- クラス内のメソッドが不十分であったものに対して追加を行った。
- 重複度を記載していない部分があったので加えた
- クラス図の配置を変更した

### 作成したもの

作成できたものは図である。

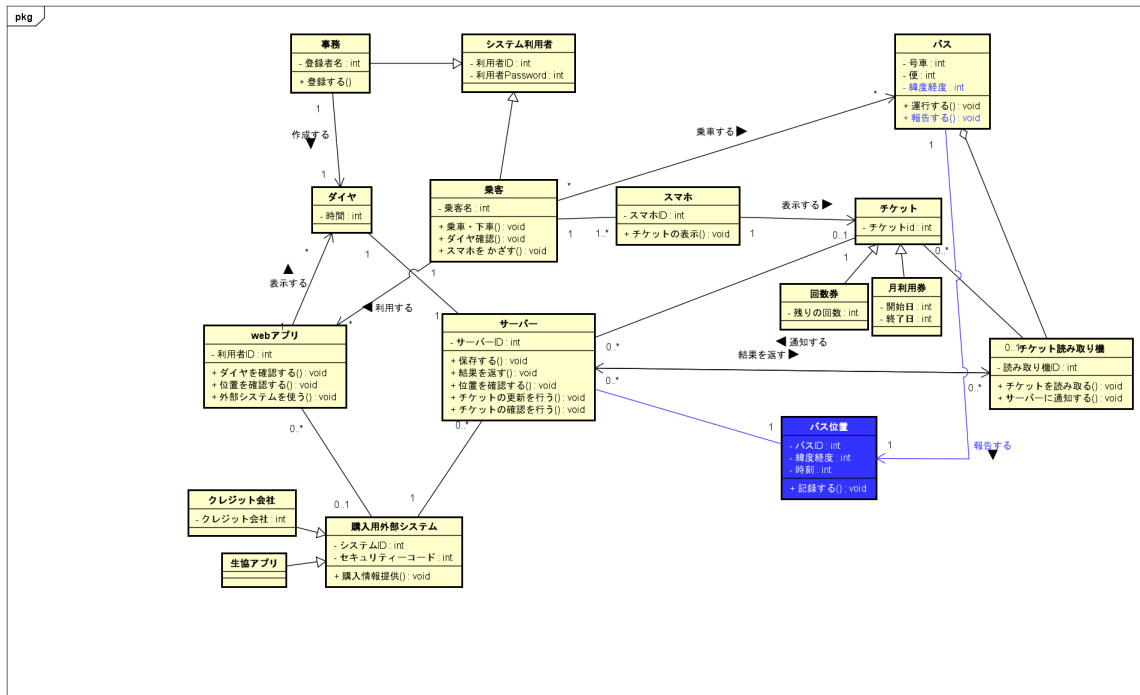


図 10 クラス図

## 6 シーケンス図

### 概要

作成したシーケンス図は乗降用、チケット購入用、のシーケンス図を作成した。基本的にユースケース記述に記載したことをそのまま実現した。これらの図にはクラス図で出てきたもののみを用いて実現するよう工夫した。要するに、新しいクラスを作るような作業は行わなかった。

### レビュー結果メモ

- ユースケース記述との相違点があったので、順番を入れ替える等の作業を行った

### 作成した図

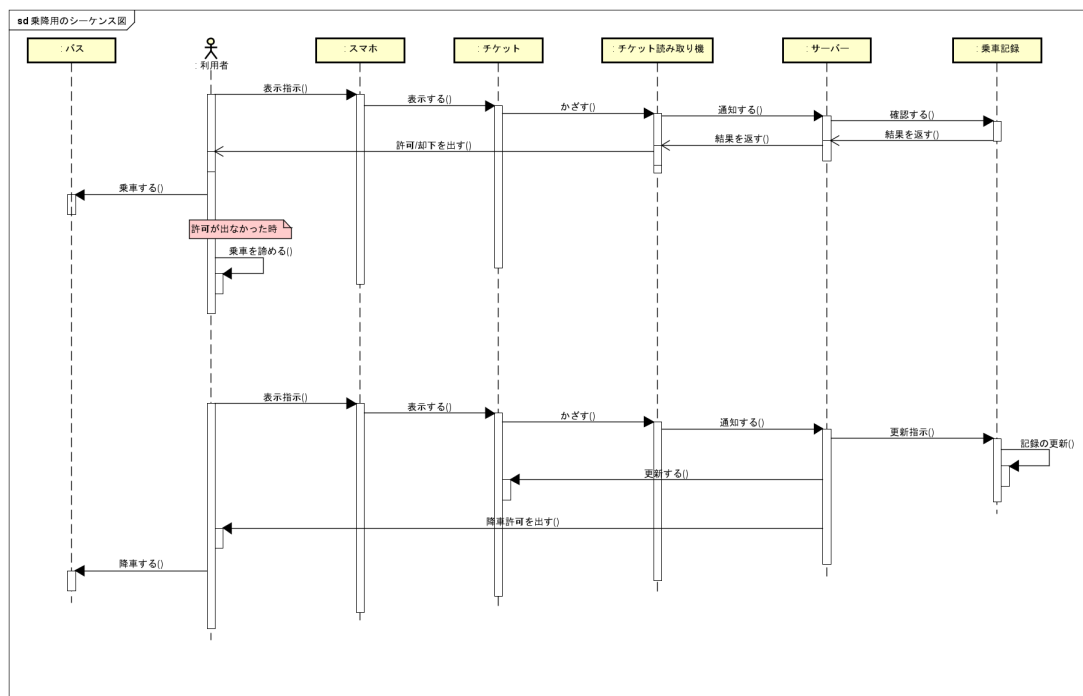


図 11 チケット購入用のシーケンス図

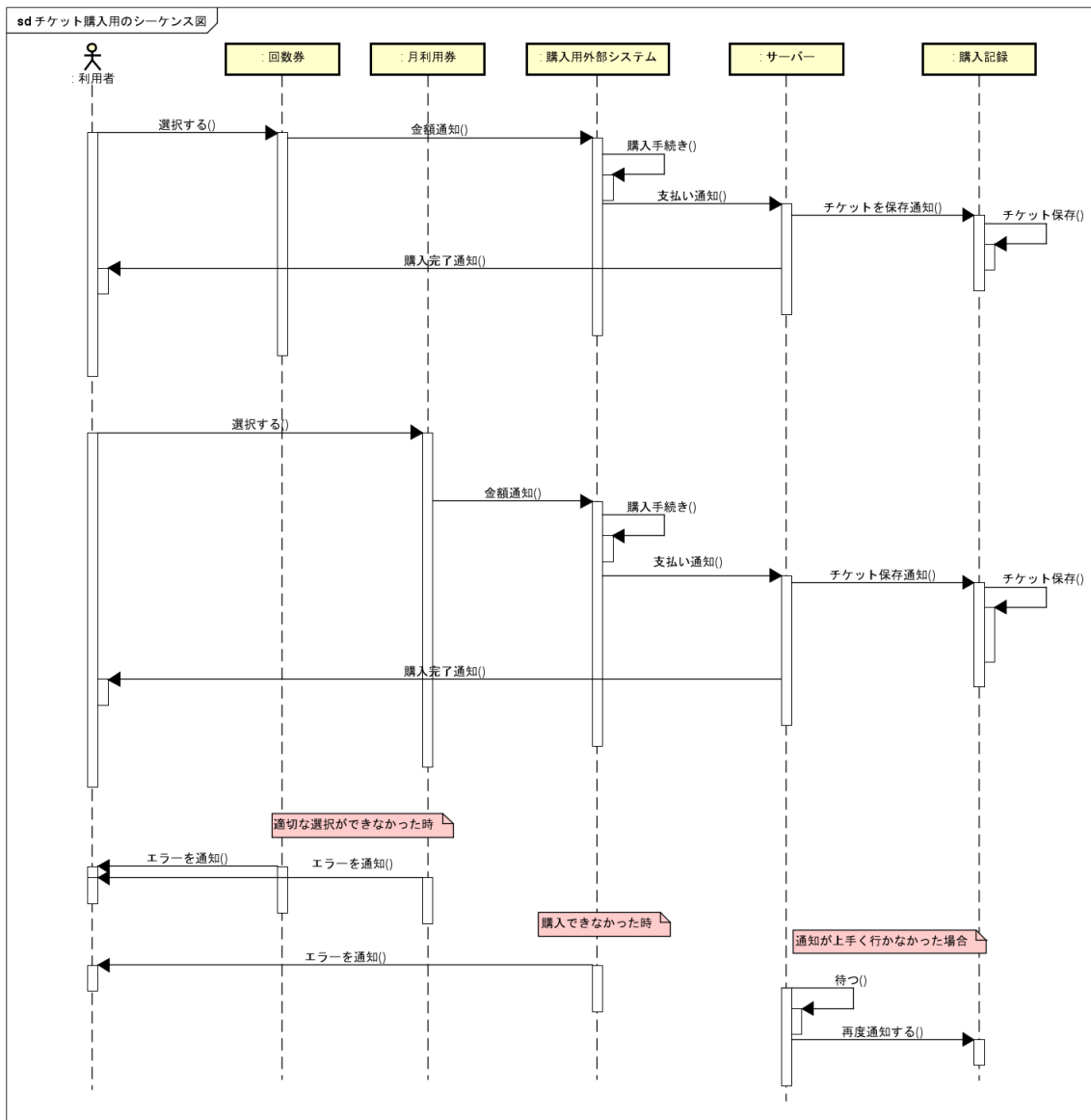


図 12 チケット購入用のシーケンス図

## 7 アクティビティ図

### 概要

作成したアクティビティ図は「バスの位置を確認する」と「バスの位置を記録する」である。

### 作成した図

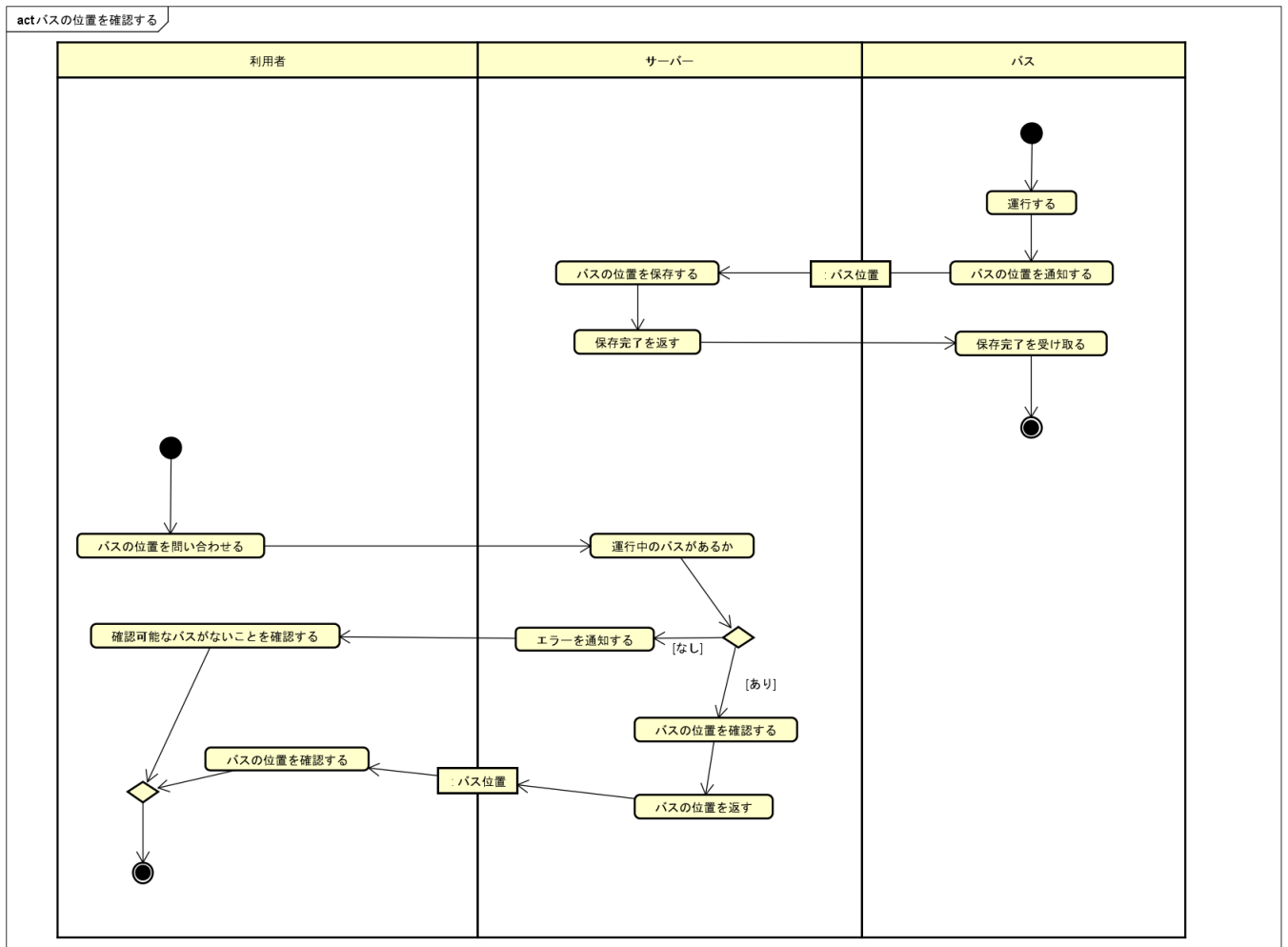


図 13 バスの位置に関するアクティビティ図

## 8 ステートマシン図

### 概要

作成したステートマシン図は「回数券」「月利用券」の2種類である。

チケットを購入すると使用待機中に遷移する。乗降を持って1回の使用が終わり、そこで残りの回数が0回か1回以上かで取りうる状態が異なるよう設計した。月利用券は、定期券と同じで使用期間が設定されているという判断をした。使用待機中から使用できるのは、使用期限以内に設定し、期限が切れると同時に「期限切れ」に状態が遷移する。

### レビュー

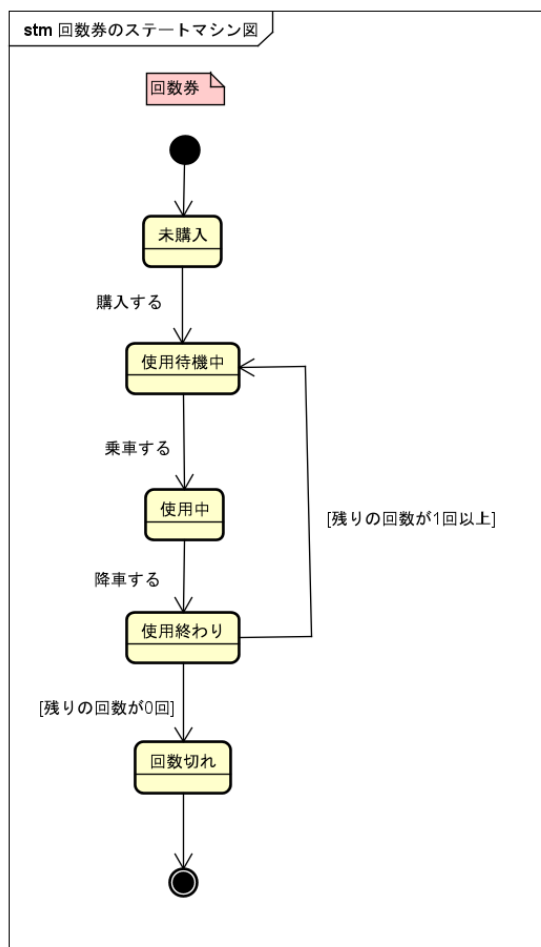


図 14 回数券のステートマシン図

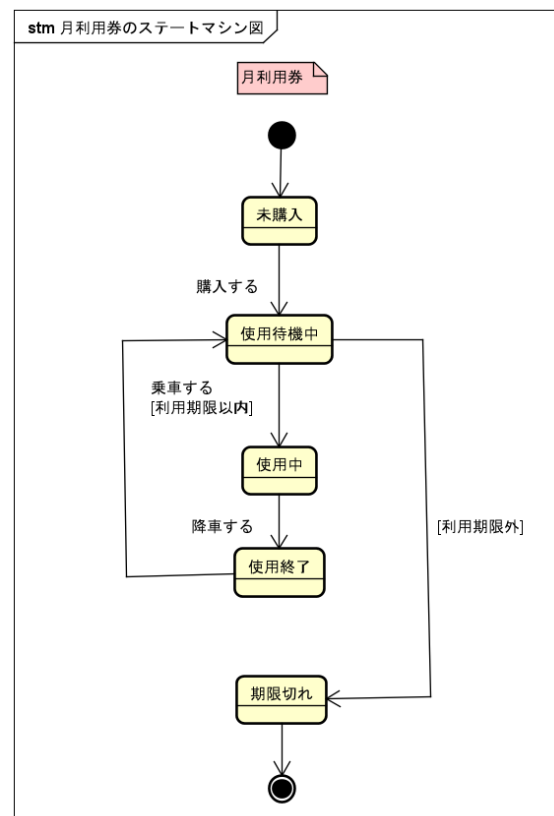


図 15 利用券のステートマシン図

## 9 まとめ

前回使用した UML 図に対して複数回のレビュー訂正を行い今回のレポート作成を行った。クラス図、シーケンス図に関してはまだ理解できていない部分があり、大きく訂正を要した。クラス図に関しては、クラスが多すぎて図がわかりにくくなってしまった。今後の作成でクラスの数意識していきたい。

レビューを図を作成する事に行ったので、後の図の作成に与える影響が少なかったのは良かった。影響が少ないというよりは、事前に作った図を元に次の図を作成したので、前の図とあっていない等のミスは減った。今後の作成においても、図を作成するごとにレビューを行い、後の作業に影響を与えないように工夫したい。