

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 20 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

これらの図を作成し、分析が一定できると、設計モデルの作成に着手した。設計モデルでは、パターンの適用を駆使し、システムがより現実的になるよう改変を加えた。

3 ユースケース図

概要

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。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「システムにログイン」、「履修登録、削除」等の直接行うユースケースを作成した。「履修を促す」というシステム側から行われるユースケースを関連付けている。

教員は「システムにログイン」、「成績の報告をする」というユースケースを作成した。また、学生と同様に成績報告に関する通知を行うユースケースを関連付けている。

作成した図

レビュー等を通して作成したユースケースが図1である。

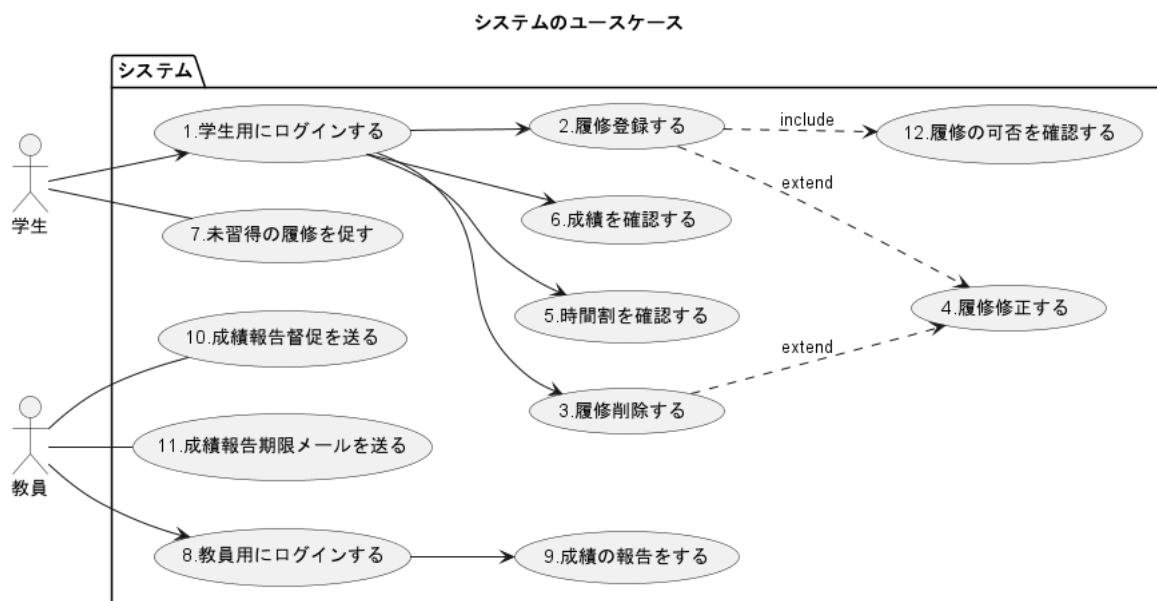


図1 ユースケース図

レビュー内容

ユースケース「履修の可否を確認する」に関して、「学生」と関連をつけた方が良いということで、「履修登録する」に include で関連付けた。また、ユースケース「ログインする」が必要だと思ったので、作成し、他のユースケースと関連をつけた。

概要から「履修修正」という要件があったが、今回のシステムでは履修削除と履修登録を組み合わせて行うと考えた。

4 ユースケース記述

概要

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。ユースケースそれぞれに対して基本系列を作成し、加えて代替系列が必要なユースケースでは適宜作成を行った。

作成した図

1. 学生用にログインする

項目	内容
UML	1 学生用にログインする
基本系列	1.1 学生ID, Passwordを学生が入力する 1.2 IDとPasswordが確認できたら個人ページにログインする
代替系列	1.2E ID, Passwordが違ったらエラーを出す

2. 履修登録する

項目	内容
UML	2 履修登録する
基本系列	2.1 学生が曜日, 時間を選択する 2.2 システムがその学期で学生に適切な科目を一覧として表示する 2.3 学生はその一覧から科目を選択する 2.4 学生はその科目を履修登録する 2.5 システムはその科目に学生を登録する
代替系列	2.4E その科目がすでに履修済みであったり, 他の科目と時間が被っていたら登録できない旨を通知する

3. 履修削除する

項目	内容
UML	3 履修削除する
基本系列	3.1 学生が曜日, 時間を選択する 3.2 システムが科目を表示する 3.3 学生はその一覧から削除したい科目を選択する 3.4 システムはその科目の履修を削除する
代替系列	3.4E その科目の履修削除期間外だった場合は削除できない旨を通知する

4. 履修修正する

項目	内容
UML	4 履修修正する
基本系列	4.1 まず履修削除で対応する科目の削除を行う 4.2 その曜日, 時間で履修登録をする(その後の動作は履修登録と同じ)
代替系列	

5. 時間割を確認する

項目	内容
UML	5 時間割を確認する
基本系列	5.1 学生は時間割オプションを選択する 5.2 学生は学期を選択する 5.2 システムは学期でその学生が履修している科目を基に時間割を生成する
代替系列	

6. 成績を確認する

項目	内容
UML	6 成績を確認する
基本系列	6.1 学生は成績オプションを選択する 6.2 システムはその学生の履修している科目とその成績を一覧として表示する
代替系列	

7. 未習得の履修を促す

項目	内容
UML	7 未習得の履修を促す
基本系列	7.1 システムはその学生に必要な科目が習得されているか確認する 7.2 習得されていない場合、学生にメールで通知する
代替系列	

8. 教員用にログインする

項目	内容
UML	8 教員用にログインする
基本系列	8.1 教員ID, Passwordを教員が入力する 8.2 IDとPasswordが確認できたら個人ページにログインする
代替系列	8.2E ID, Passwordが違ったらエラーを出す

9. 成績の報告をする

項目	内容
UML	9 成績を報告する
基本系列	9.1 成績報告オプションを選択する 9.2 システムは教員が担当している科目一覧を表示する 9.3 教員は成績入力する科目を選択する 9.4 システムはその科目の履修者一覧を表示する 9.5 教員は履修者それぞれに点数を入力していく 9.6 教員が保存を押すと、システムはその成績を保存すると同時に、仮成績として登録する 9.7 教員が提出と押すと、システムはその成績を保存すると同時に、本成績として登録する
代替系列	9.7E すべての履修者に成績が入力されていなかった場合提出付加の旨を通知する

10. 成績報告督促を送る

項目	内容
UML	10 成績報告督促を送る
基本系列	10.1 システムは成績報告期限になると、それぞれの科目に対して成績報告ステータスが提出になっているかを確認する 10.2 システムは提出になっていない科目の担当教員に対して成績報告督促メールと送信する
代替系列	

11. 成績報告期限メールを送る

項目	内容
UML	11 成績報告期限メールを送る
基本系列	11.1 システムは成績報告期限の一週間前に、それぞれの科目に対して成績報告ステータスが提出になっているかを確認する 11.2 システムは提出になっていない科目の担当教員に対して期限を通知するメールを送信する
代替系列	

12. 履修の可否を確認する

項目	内容
UML	12 履修の可否を確認する
基本系列	12.1 学生が履修登録しようとしている科目がすでに履修済みであるか、時間が被っていないかを確認し、その結果を通知する
代替系列	

5 クラス図

概要

全体のクラス図は図2である。この図に対して、学生、教員それぞれに関係するクラスで分割して物で詳細な関係を分析した。そのため、2における関係に重複度が内藤の不具合があるが、それは分割した詳細な図で行っているため問題ないと考えている。

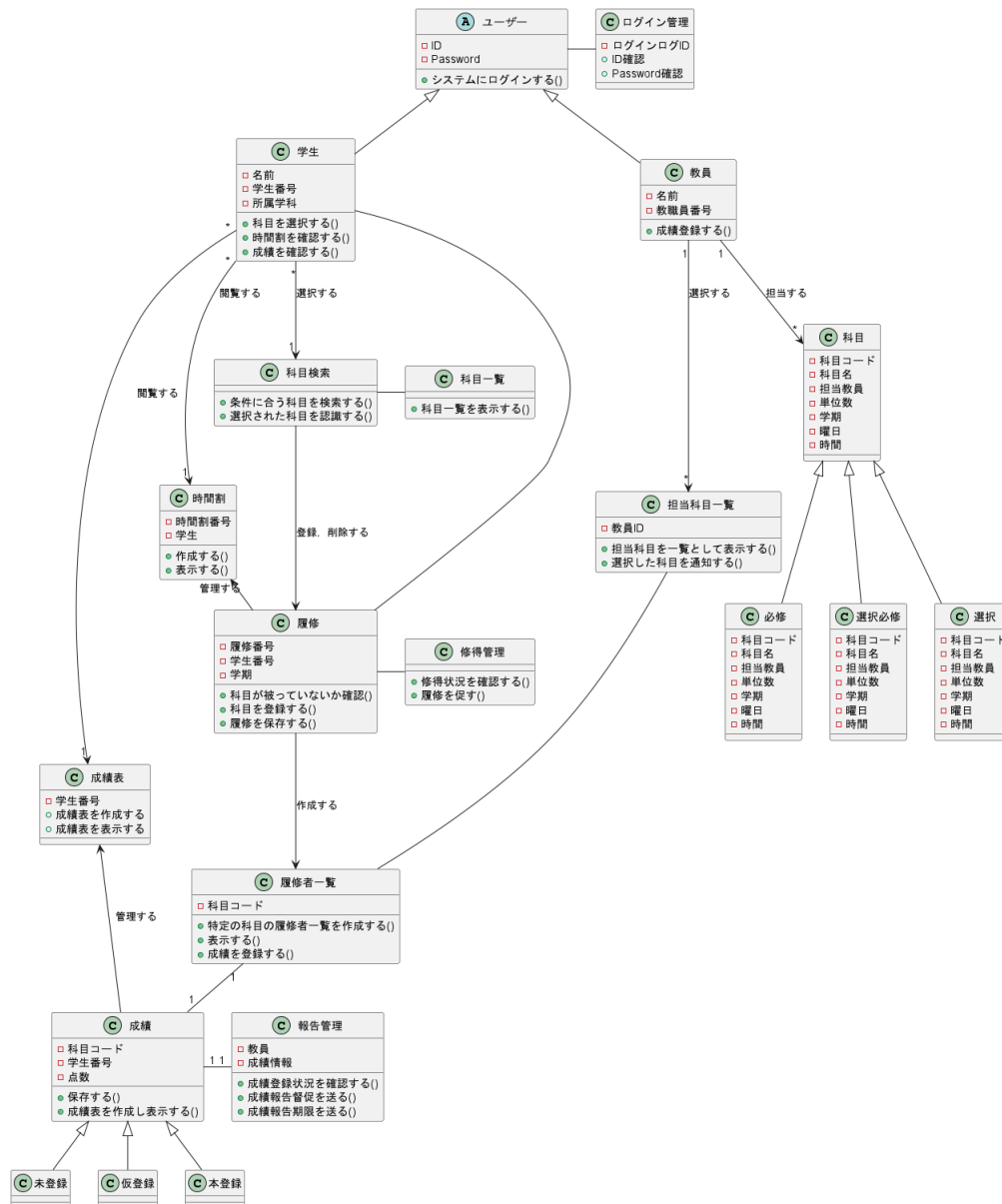


図2 クラス図

ユーザー

図3ではユーザーと、それを管理するための部分をまとめたものである。

■学生クラス 学生の名前や学生番号が属性として保存されている。科目の選択や時間割を確認等の動作を行う。

■**教員クラス** 教員の名前や教員 ID が属性として保存されている。成績の登録という動作を行う。

■**ユーザークラス** 学生と教員のスーパークラスであり、システムのログインというメソッドを共通化して持っている。

■**ログイン管理クラス** ユーザーがログインした際にログを取る等の動作を行うクラスである。

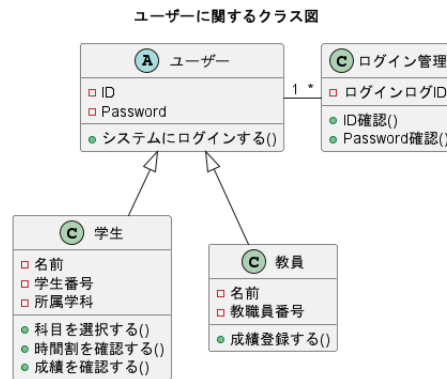


図 3 ユーザーに関するクラス図

学生

図 4 では、学生に関する部分をまとめたクラス図である。

■**時間割** 学生毎に時間割が作成される。履修クラスを参考にしながら時間割を作成し、時間割を表示するという動作を行う。

■**科目検索と科目一覧クラス** 学生が科目を検索する際に行うクラスである。学生が曜日と時間を指定すると科目一覧クラスと連携して条件に合う科目を一覧として表示する。

■**履修クラス** 履修クラスはある学生がどの科目を履修しているかを表すクラスである。学生毎に作成されることを想定しているまた、履修登録、削除ができるかも確認をする。このクラスで時間割を管理することを想定している。成績クラスを参照し、成績も一緒に保存することを想定する。

■**修得管理クラス** 履修クラスで履修状況を確認し、修得できていなかったら修得を促すクラスである。

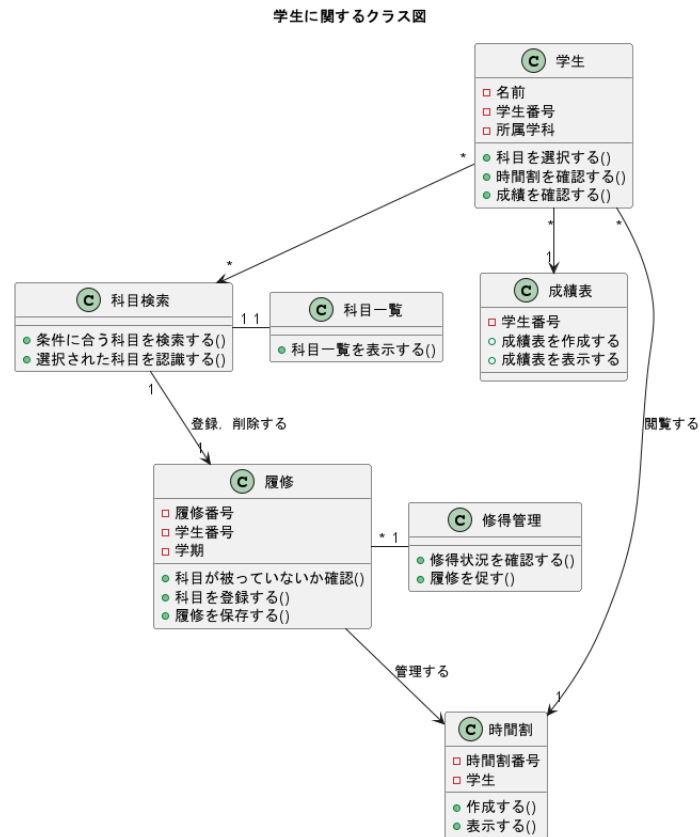


図 4 学生に関するクラス図

教員

図 5 では、教員に関する部分をまとめたクラス図である。

■**科目クラス** 教員は複数の科目を担当することから、教員と関連をつけている。このクラスは学生と直接かわりがないが、科目一覧等で間接的に関連付けることで履修等の情報を取れるようにしている。また、科目には必修、選択必修、選択の 3 種類の分類がされ、学生毎に科目の立ち位置も変わることから属性に追加するようにした。

■**担当科目一覧クラス** 教員が担当している科目を一覧として表示するクラスである。一人の教員に対して一つの担当科目一覧があることから 1 対 1 対応をしている。

■**履修者一覧クラス** 担当科目一覧で受け取った科目の履修者一覧を表示するためのクラスである。このクラスは履修クラスから作成、管理されることを想定している。

■**成績関連クラス** 成績クラスは学生、科目毎にオブジェクトが作成されることを想定したクラス図である。このクラスを基に成績表クラスが作成、管理されている。また、成績クラスには、未登録、仮登録、本登録という教員の成績登録状況を持っている。その状況を基に教員に通知を行う報告管理クラスが関連づいている。

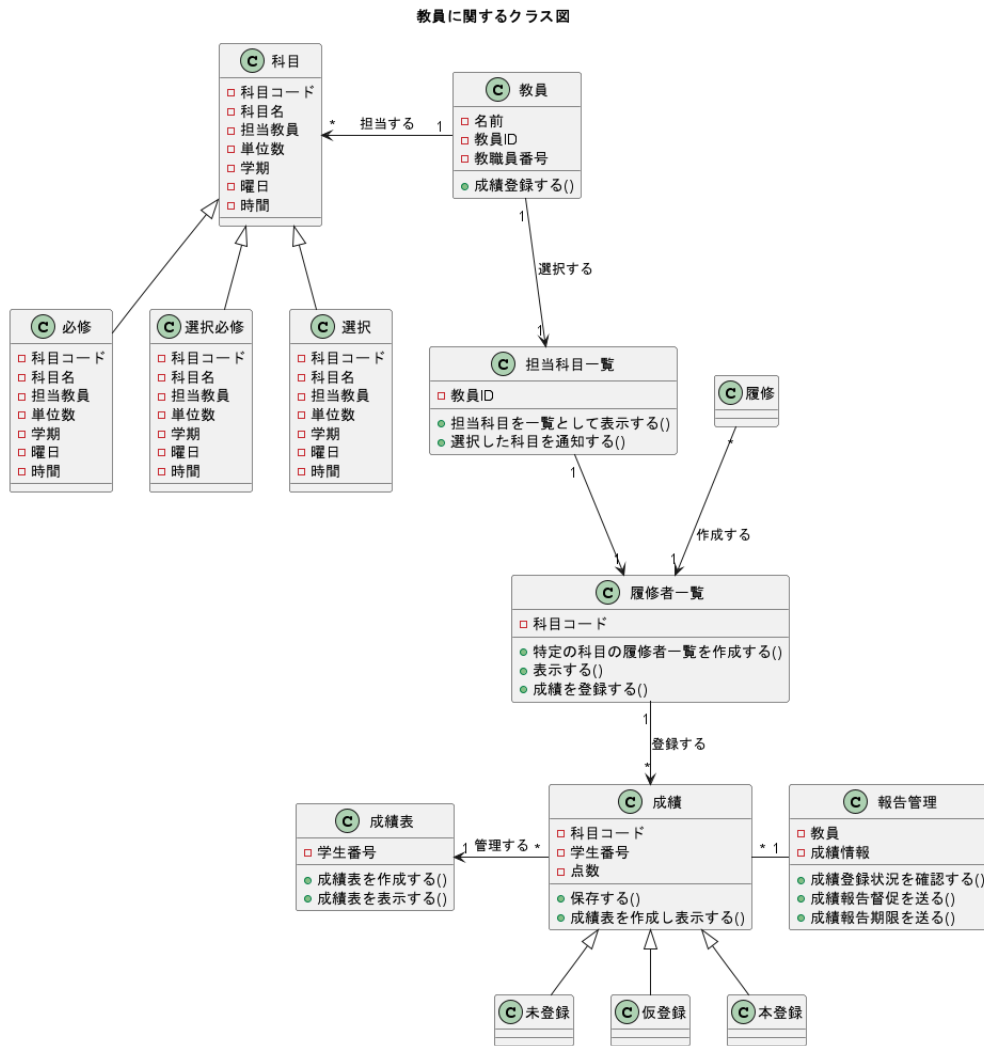


図5 教員に関するクラス図

レビュー内容

- クラス図間の関連が複雑で、必要のない部分まで関連付けていたためかなりの省略を加えた。
- 重複度が登録されていない部分があったので加えた。
- 履修クラスに対して多くのメソッドがあり、責任が大きすぎたので他のクラスを作成することで責任をできるだけ小さくした。
- 教員の成績登録の方法がシーケンス図等を通して変更されたことから、クラス図にも適用した。
- 科目に必修等の分類があることから継承したクラスを新たに作成した。

6 シーケンス図

概要

作成したシーケンス図は、

- 図 6: ログイン時のシーケンス図
- 図 7: 履修登録，削除のシーケンス図
- 図 8: 時間割確認のシーケンス図
- 図 9: 修得を促すのシーケンス図
- 図 10: 成績を確認，登録のシーケンス図
- 図 11: 成績報告メールのシーケンス図

である．

作成した図

ログイン

ユーザーという学生，教員ともにログインを行う．ID と Password をユーザーが確認し，ログイン管理が確認を行っている．

履修登録，削除

履修登録，削除をまとめて行うように工夫を行った．学生が選択した科目が履修登録されていなかった場合は履修登録をし，されていた場合は削除するというシーケンスにした．ユースケースやシステム概要には履修修正というものがあったが，今回のシステムは履修削除と登録を組み合わせる行うことからシーケンス図の作成は省略した．

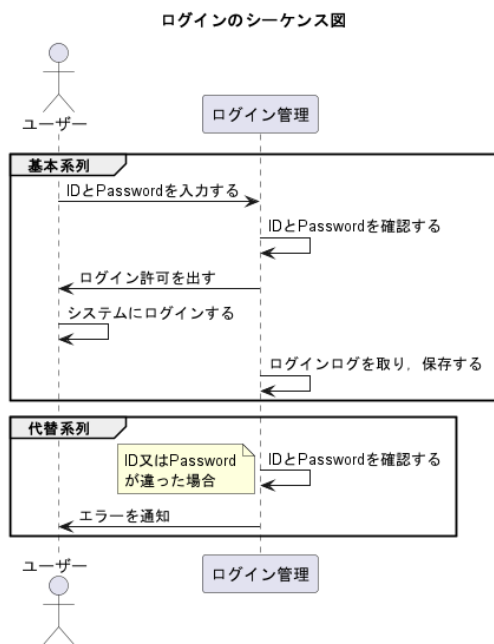


図6 ログイン時のシーケンス図

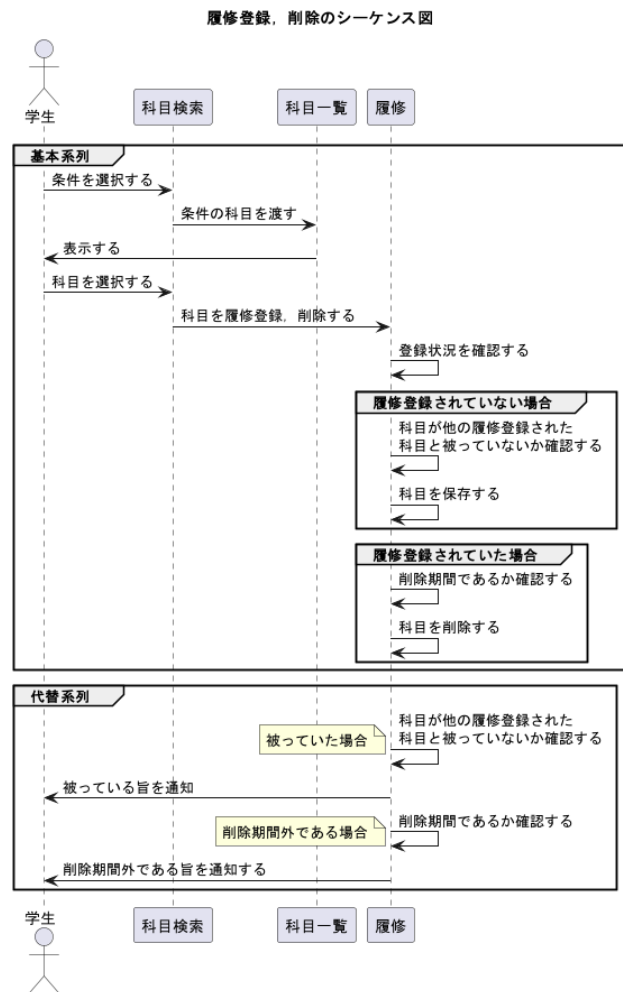


図7 履修登録、履修削除のシーケンス図

時間割

履修クラスを反映し時間割を作成することから、シーケンス図を作成した。システム側は事前に時間割を作成しており、学生はそれを確認するのみと想定している。

履修を促す

システム側が履修と必修を確認し、必修が修得されていなかった場合に修得を促す通知を出すと想定している。

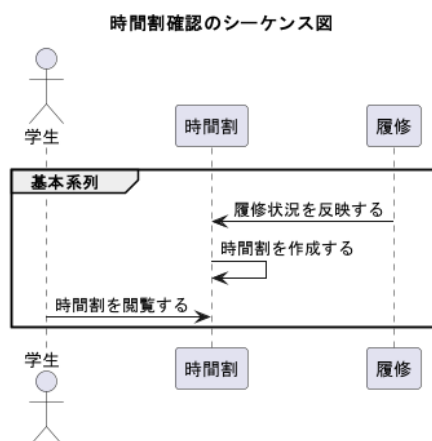


図 8 時間割確認のシーケンス図

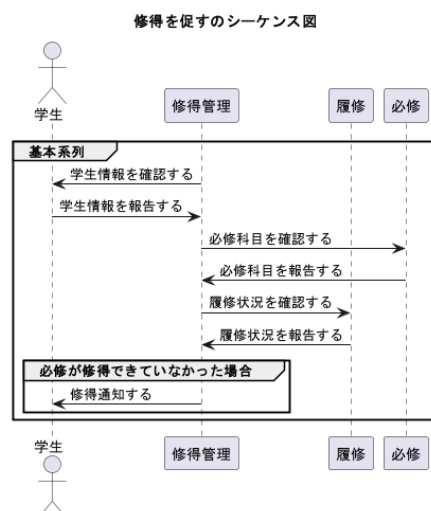


図 9 修得を促すシーケンス図

成績

教員が担当科目一覧から科目を選択し、その科目の履修者一覧をシステムが表示する。そこで教員が成績を入力し、システムが保存をしている。その保存された成績を基に成績表をシステムが作成し、学生はそれを閲覧するのみである。

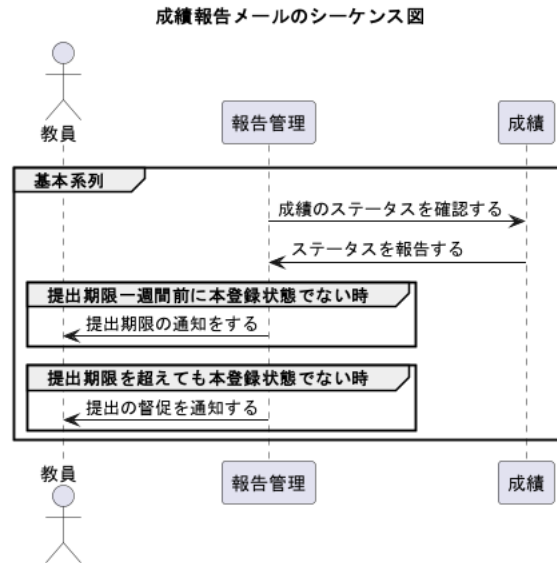


図 11 成績報告メールのシーケンス図

レビュー内容

- 履修登録、削除を同じシーケンスで表せるような動作にするよう工夫を行った。
- 時間割や成績確認では、システムが作成し、学生が確認できる状態に変更した。
- 複数のシーケンス図では流れが変だったので変更を加えた。

7 アクティビティ図

概要

作成したアクティビティ図は,

- 図 12: ログインするのアクティビティ図
- 図 13: 履修登録, 削除のアクティビティ図
- 図 14: 時間割を確認のアクティビティ図
- 図 15: 成績を確認のアクティビティ図
- 図 16: 成績を登録するのアクティビティ図
- 図 17: 未修得の履修を促すのアクティビティ図
- 図 18: 成績報告の通知を出す

である.

作成した図

ログイン

今回は学生のみで作成しているが, 教員でも同じ動作をする. 学生が入力した ID, Password をシステムが正しいか確認し, それぞれで動作が異なる.

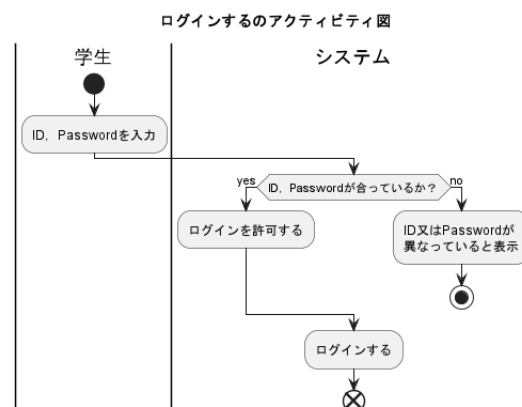


図 12 ログインするのアクティビティ図

■履修登録, 削除 学生が曜日, 時間を選択し, システムが条件にあう科目を表示する. 学生がその科目を選択することで, その科目の状況によって動作が異なる. それぞれ登録や削除ができるかを確認し, エラーの場合はエラーを表示する動作を行う.

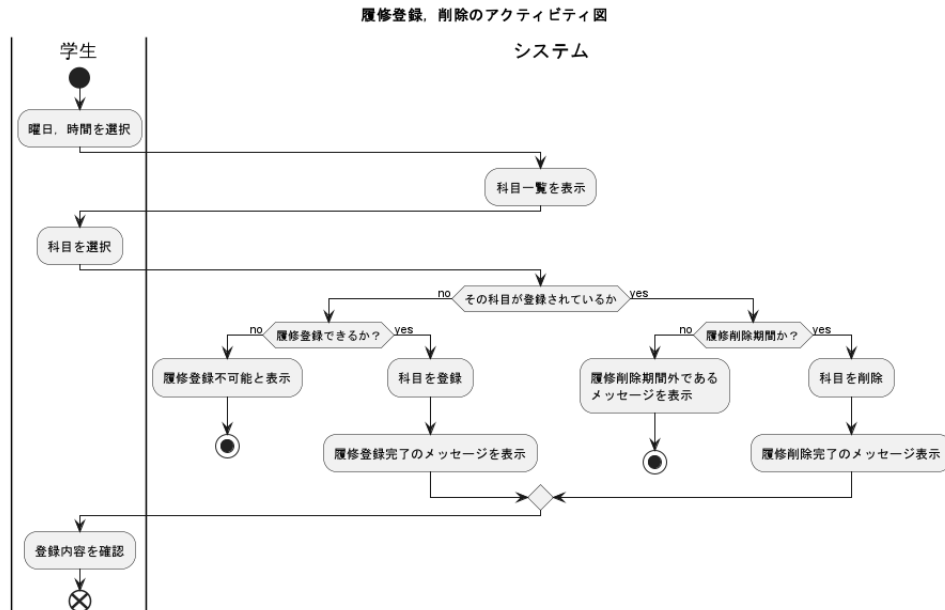


図 13 履修登録、削除のアクティビティ図のアクティビティ図

時間割、成績

システムが時間割や成績を作成し、学生がそれを確認している。

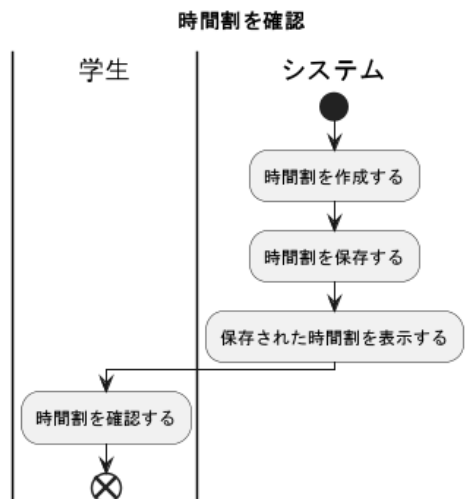


図 14 時間割を確認するのアクティビティ図

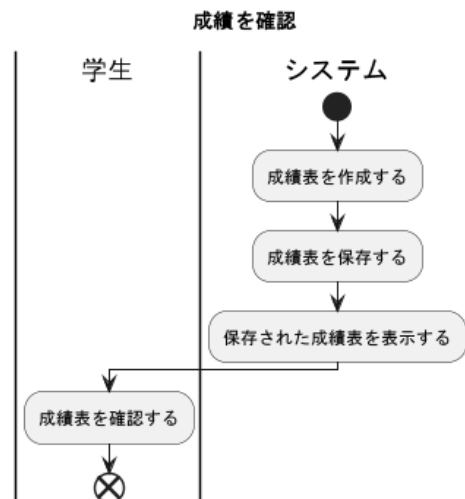


図 15 成績を確認するのアクティビティ図

成績を登録

教員が担当している科目を表示、教員が科目を選択する。システムはその科目の履修者を表示し、教員がせ成績を入力する。そのまま教員が保存すると仮成績として保存し、すべての履修者に成績が入力されていた場合、本成績として保存する。

未習得の履修

システムが必修科目や履修状況を確認することで、修得したほうが良いと判断した際に学生に修得を促すようにしている。

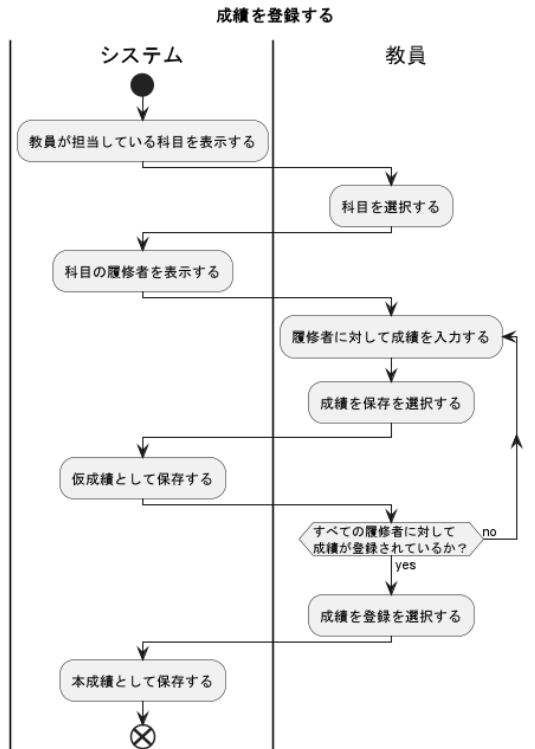


図 16 成績を登録するのアクティビティ図

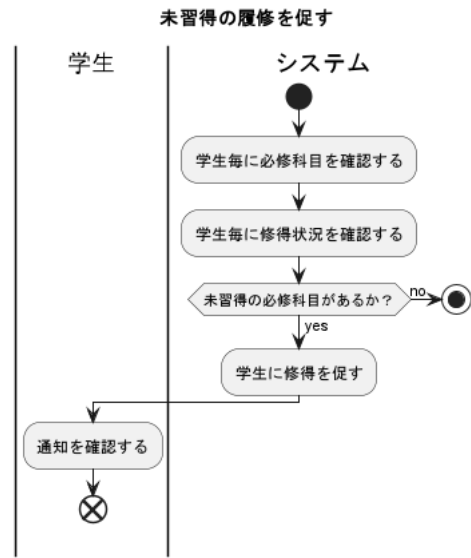


図 17 未習得の履修を促すのアクティビティ図

成績報告

成績報告の期間によって動作が異なる。すべての履修者に対して成績が登録されていた場合は何も行わないが、登録されていない場合は成績報告の督促等を通知する。

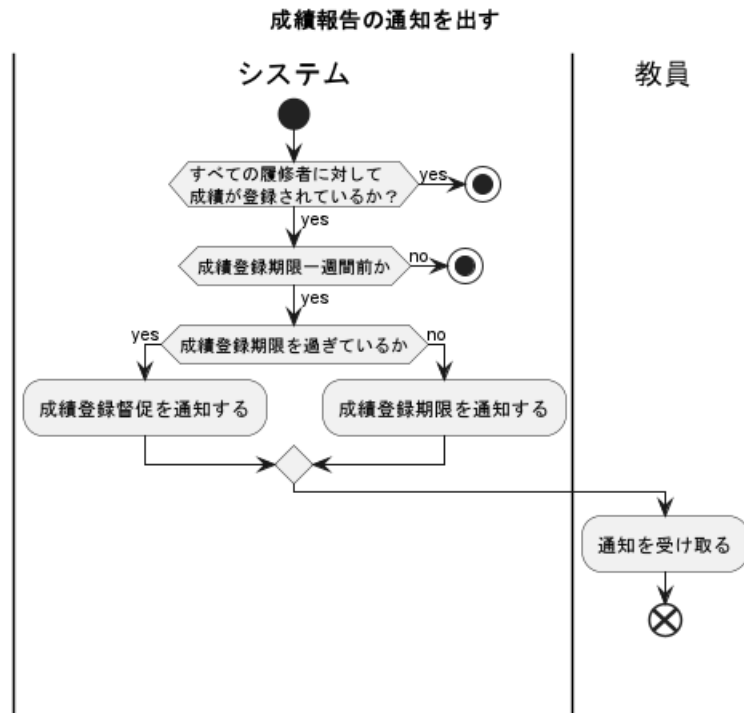


図 18 成績報告の通知を出すアクティビティ図

8 ステートマシン図

概要

今回のシステムにおいて、「成績」の報告に関してステートマシン図を作成した。

作成した図

成績は、「未登録状態」→「仮成績」→「本成績」の順で変化する。その図を図 19 で示す。

成績のステートマシン図

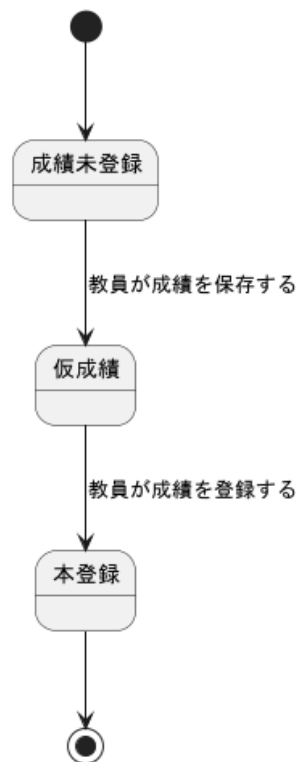


図 19 成績のステートマシン図

9 設計モデルのクラス図

作成したクラス図に対して、パターンを導入する等の設計モデルの作成を行った。また、シーケンス図やステートマシン図等を通して追加したほうが良いと考えたクラス図も追加する作業を行った。

ユーザー

図3のユーザーに関するクラス図に対しては、Observerパターンを使用し、ログインの履歴を取る「ログイン管理」クラスがログインを適切に管理できるよう改変を行った。

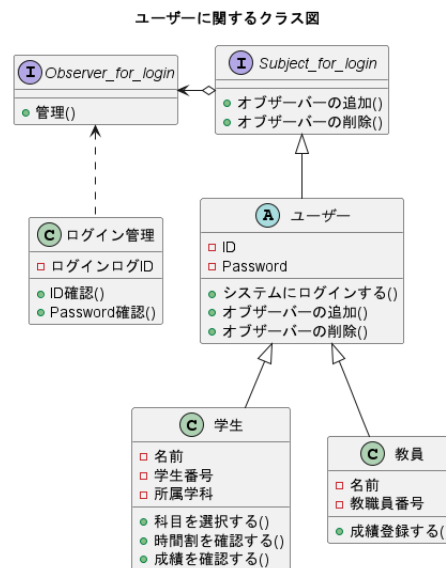


図 20 設計モデルを適用したログインに関するクラス図

学生

図4の学生に関するクラス図に関しては、ObserverパターンとStateパターンを用いて設計モデルを作成した。Observerパターンにおいては、学生の習得状況を取得する際に適用を行った。Stateパターンは当初のクラス図では使用していなかった、修得状況に対して適用を行った。修得状況は「必修科目」クラスに関連付けられており、学生それぞれの修得状況によって状態が変更される。

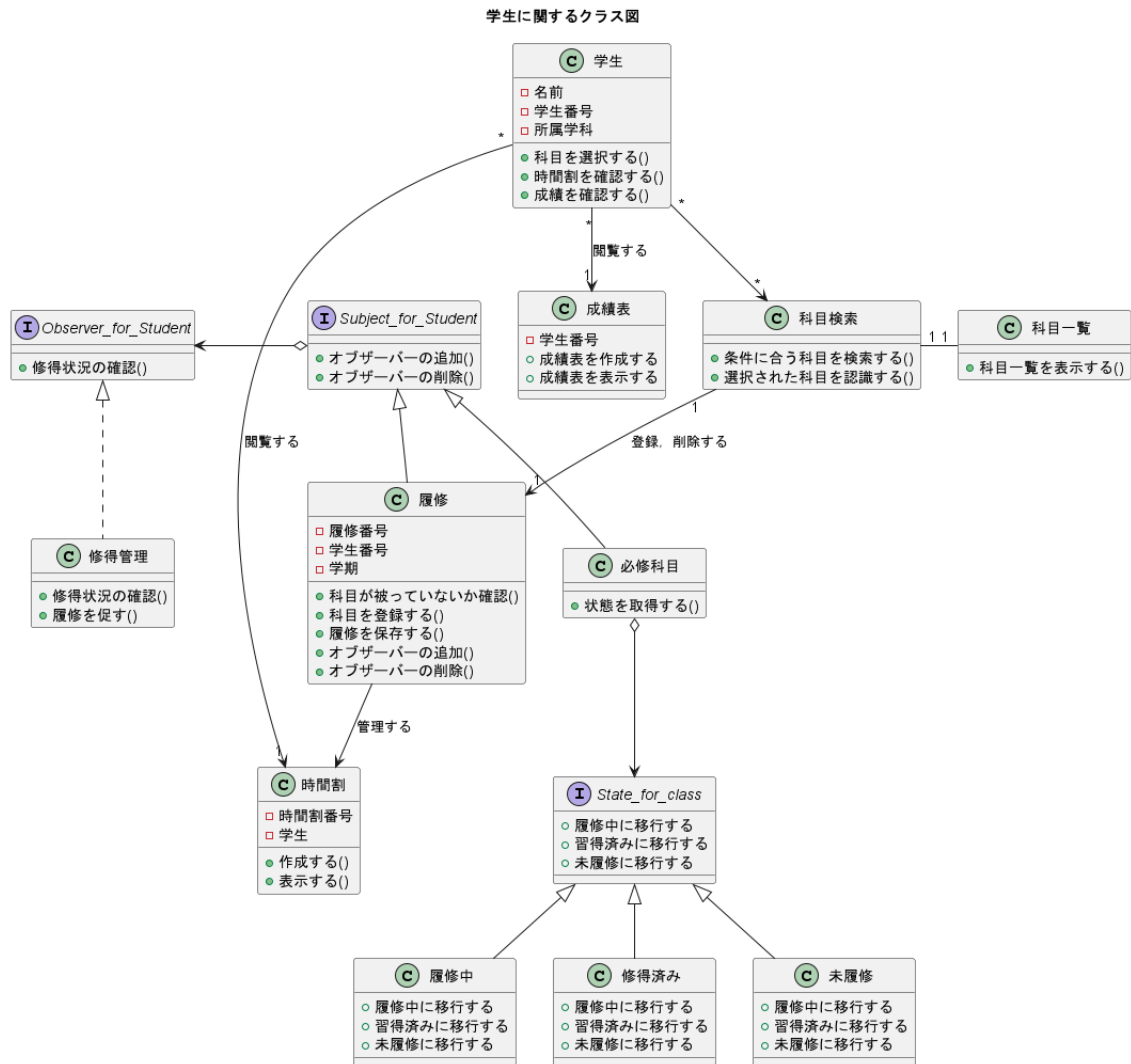


図 21 設計モデルを適用した学生に関するクラス図

教員

図 5 の教員に関するクラス図に対しても、Observer パターンと State パターンを用いた。Observer パターンにおいては、教員が成績を報告しているかの管理に対して使用した。State パターンにおいても、教員が成績の報告状況にたいして使用している。

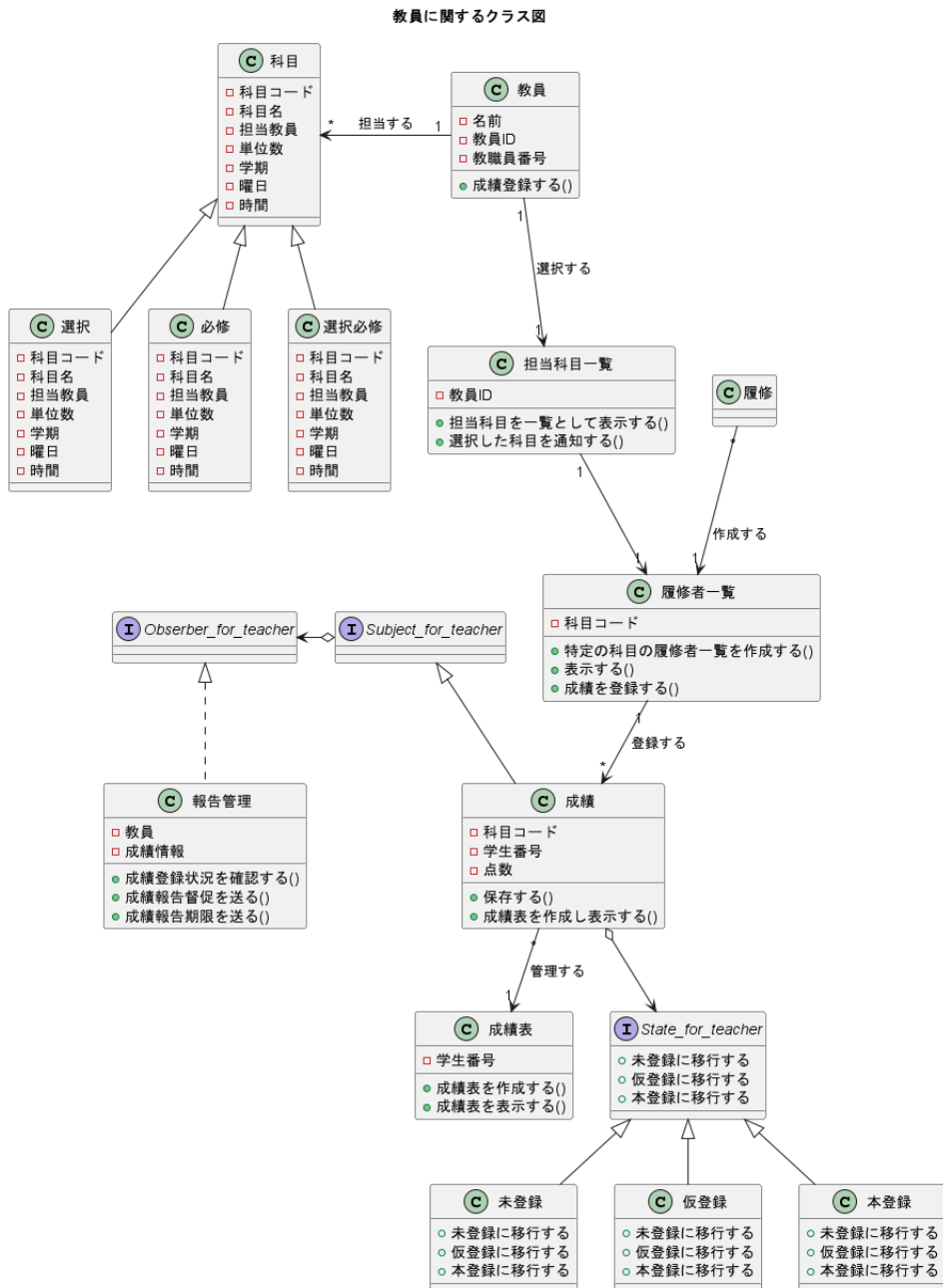


図 22 設計モデルを適用した教員に関するクラス図