教室予約システム

システム構造設計書

プログラム構造設計書

2023年6月26日　第3.1版

2023年6月22日　第3.0版

2023年6月15日　第2.1版

2023年4月02日　第2.0版

2023年4月01日　第1.1版

2023年3月27日　第1.0版

2023年3月24日　第0.0版

東京国際工科専門職大学

工科学部 情報工学科

黒羽　　　晟

佐藤　礼央奈

小林　洋二朗

森田　　　輪

秦　　志於里

はじめに

　本ドキュメントは，東京国際工科専門職大学内の教室予約システムに関するソフトウェアシステム構造及びプログラム構造について規定するものである．

なお，本システムは，スパイラルにて開発を行うが，イテレーション毎に本ドキュメントを更新するものとする．ベースとなるユーザインタフェース仕様書の版数については，変更履歴に明記する．

【本プロジェクトにおける工程と成果物】

システムテスト

仕様書／成績書

ユーザインタフェース設計

ユーザインタフェース仕様書

システムテスト

仕様書／成績書

システム構造設計／

プログラム構造設計

システム構造設計書／

プログラム構造設計書

本ドキュメントの位置付け

プログラミング

ソースプログラム

変更履歴

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | 版数 | 概要 | 頁 | 変更者 | 変更日 | |
| 1 | 0.0 | ログイン・ログアウト対応版．レビュー直前版発行 | 全頁 | 荒金匡徳 | 2023/3/24 | |
| 2 | 1.0 | レビュー完了のため正式版としてリリース．  ログイン・ログアウト機能対応版ここまで | 全頁 | 荒金匡徳 | 2023/3/27 | |
| 3 | 1.1 | 教室概要表示機能対応．レビュー直前版発行 | 全頁 | 荒金匡徳 | 2023/4/1 | |
|  |  | 2. 1 MainFrameの構成を変更，panelNorthの分割とpanel上への部品配置について説明追記 | 2.1 |  |  | |
|  |  | 3. クラス図にChoiceFacilityを追加 | 3. |  |  | |
|  |  | 3.2.MainFrameの追加メソッド・処理追記 | 3.2 |  |  | |
|  |  | 3.3.ReservationControlのメソッド追記 | 3.3 |  |  | |
|  |  | 3.5.ChoiceFacility追記 | 3.5 |  |  | |
|  |  | 4. シーケンスでMainFrameより起動される機能が増えるため，シーケンスを細分化 | 4 |  |  | |
| 4 | 2.0 | レビュー完了のため正式版としてリリース．  教室概要表示対応版ここまで | 全頁 | 荒金匡徳 | 2022/4/24 | |
| 5 | 2.1 | 3.6 ReservationDialog Class　メソッド追記 | 3.6 | 佐藤礼央奈 | 2023/6/15 | |
|  |  | 3.7 ChoiceHour Class メソッド追記 | 3.7 | 小林洋二朗 | 2023/6/15 | |
|  |  | 3.8 ChoiceMinute Class メソッド追記 | 3.8 | 小林洋二朗 | 2023/6/15 | |
|  |  | 全体クラス図の追加・変更 | 3. | 佐藤礼央奈 | 2023/6/15 | |
|  |  | 細分化したシーケンス追加・変更 | 4. | 岡村昂弥  黒羽晟 | 2023/6/15 | |
| 6 | 3.0 | 3.9 ReservationStatusDialog class メソッド追記 | 3.9 | 小林洋二朗 | 2023/6/22 | |
|  |  | 3.2 mainFrame メソッド追記 | 3.2 | 小林洋二朗 | 2023/6/22 | |
|  |  | 3.3 ResrvationControl メソッド追記 | 3.3 | 小林洋二朗 | 2023/6/22 | |
|  |  | クラス図の追記 | 3 | 小林洋二朗 | 2023/6/22 | |
| 7 | 3.1 | 細分化したシーケンス追加 | 4.5 | 小林洋二朗 | 2023/6/26 | |
| 8 | 3.2 | 2.3 Reservation Dialog window画面 画像説明追加 | 2.3 | 佐藤礼央奈 | 2023/6/29 | |
|  |  | 2.4 Reservation Status Dialog window画面 画像説明追加 | 2.4 | 佐藤礼央奈 | 2023/6/29 | |
|  |  | 2.2 Login Dialog 配置部品追記 | 2.2 | 佐藤礼央奈 | 2023/6/29 | |
|  |  | 2.3 Reservation Dialog 配置部品追記 | 2.3 | 佐藤礼央奈 | 2023/6/29 | |
|  |  | 2.4 Reservation Status Dialog配置部品追記 | 2.4 | 佐藤礼央奈 | 2023/6/29 | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  | |
|  |  |  |  |  |  |

Table of Contents

[1 基本アーキテクチャ 4](#_Toc138931238)

[2 Window構成 5](#_Toc138931239)

[2.1 MainFrame 5](#_Toc138931240)

[2.2 ログインダイアログ 5](#_Toc138931241)

[2.3 Reservation Dialog 6](#_Toc138931242)

[2.4 Reservation Status Dialog 6](#_Toc138931243)

[3 クラス図及び各クラス内の変数及びメソッド 7](#_Toc138931244)

[3.1 ReservationSystem 7](#_Toc138931245)

[3.1.1 mainメソッド 7](#_Toc138931246)

[3.2 MainFrame 8](#_Toc138931247)

[3.2.1 MainFrameコンストラクタ 8](#_Toc138931248)

[3.2.2 windowClosingメソッド 8](#_Toc138931249)

[3.3 ReservationControl 9](#_Toc138931250)

[3.3.1 sqlUserID 9](#_Toc138931251)

[3.3.2 sqlPassword 9](#_Toc138931252)

[3.3.3 reservationUserID 9](#_Toc138931253)

[3.3.4 flagLogin 9](#_Toc138931254)

[3.3.5 ReservationControlコンストラクタ 10](#_Toc138931255)

[3.3.6 connectDBメソッド 10](#_Toc138931256)

[3.3.7 closeDBメソッド 10](#_Toc138931257)

[3.3.8 loginLogoutメソッド 10](#_Toc138931258)

[3.3.9 getFacilityExplanationメソッド 11](#_Toc138931259)

[3.3.10 makeReservationstatusメソッド 11](#_Toc138931260)

[3.4 LoginDialog 11](#_Toc138931261)

[3.4.1 canceled 12](#_Toc138931262)

[3.4.2 LoginDialogコンストラクタ 12](#_Toc138931263)

[3.4.3 windowClosingメソッド 12](#_Toc138931264)

[3.4.4 actionPerformedメソッド 12](#_Toc138931265)

[3.5 ChoiceFacility 13](#_Toc138931266)

[3.5.1 ChoiceFacilityコンストラクタ 13](#_Toc138931267)

[3.6 ReservationDialog 13](#_Toc138931268)

[3.6.1 ReservationControl コンストラクタ 14](#_Toc138931269)

[3.6.2 **resetTimeRange(String facility)　メソッド** 14](#_Toc138931270)

[3.6.3 **itemStateChanged(ItemEvent)　メソッド** 14](#_Toc138931271)

[3.6.4 **windowClosing(WindowEvent) メソッド** 14](#_Toc138931272)

[3.6.5 actionPerformed(ActionEvent) メソッド 14](#_Toc138931273)

[3.6.6 使用されてないメソッド 15](#_Toc138931274)

[3.7 ChoiceHour Class 15](#_Toc138931275)

[3.7.1 ChoiceHourコンストラクタ 15](#_Toc138931276)

[3.7.2 resetRangeメソッド 15](#_Toc138931277)

[3.7.3 getFirstメソッド 15](#_Toc138931278)

[3.7.4 getLastメソッド 15](#_Toc138931279)

[3.7.5 getItemメソッド 15](#_Toc138931280)

[3.7.6 getItemCountメソッド 15](#_Toc138931281)

[3.8 ChoiseMinute Class 15](#_Toc138931282)

[3.8.1 ChoiceMinuteコンストラクタ 15](#_Toc138931283)

[3.9 ReservationStatusDialog Class 16](#_Toc138931284)

[3.9.1 itemstateChange 16](#_Toc138931285)

[4 シーケンス図 16](#_Toc138931286)

[4.1起動 17](#_Toc138931287)

[4.2ログイン機能 17](#_Toc138931288)

[4.2.1ログインダイアログ表示 17](#_Toc138931289)

[4.2.2ID，パスワード入力～認証完了 17](#_Toc138931290)

[4.2.3 ID，パスワード入力～ユーザID未登録 18](#_Toc138931291)

[4.2.4 ID，パスワード入力～パスワード不一致 18](#_Toc138931292)

[4.2.5認証中止 18](#_Toc138931293)

[4.3教室概要表示 18](#_Toc138931294)

[4.4新規予約（簡易版） 19](#_Toc138931295)

1. 基本アーキテクチャ

　本システムは，下図の様なMVCアーキテクチャにより構成する．

Model

View

Controller



ユーザへの情報表示

ユーザからの入力

ユーザからの入力

ユーザへの情報

処理の要求

レスポンス

MySQL

=

画面の生成，イベントの受信

　ReservationSystemクラスがViewであるMainFrameクラス及びControllerであるReservationControlクラスのインスタンスを生成する．

　MainFrameクラスはMainFrameの描画を行い，以降ユーザからの入力を受け取り，ReservationControlクラスに入力された情報を渡しながら，そのレスポンスを表示する．

ReservationControlクラスは，インスタンス生成時にModelであるMySQLとの通信を確立し，以降MainFrameからユーザからの入力を受け取ると，それに応じた処理を行い，結果をMainFrameに返す．

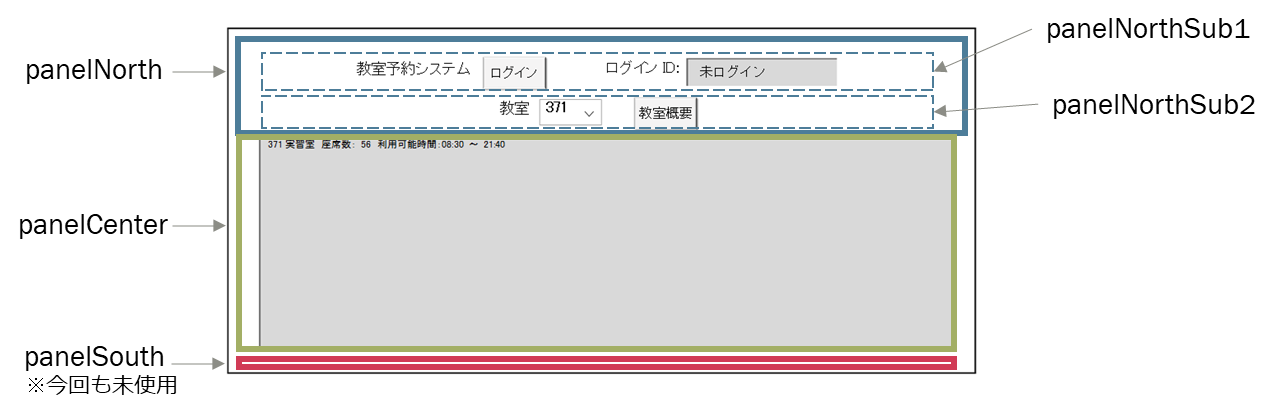
ユーザが「ログインボタン」を押下すると，MainFrameはそれをReservationControlに伝え，ReservationControlクラスでは，もう１つのViewであるLoginDialogクラスのインスタンスを生成し，以降ログイン処理が終了するまでの間，ReservationControlはLoginDialogクラスより，ユーザの入力を受け付け，処理を行う．

コンボボックスへのアイテム登録は，MainFrameクラス内で行うと煩雑になるため，choiceの派生クラスを作成し，利用する事とする．今回はChoiceFacilityクラスが該当する．

1. Window構成

本システムで使用するWindowの部品配置構成を以下に示す．

* 1. MainFrame



panelNorthにpanelNorthub1及びpanelNorthSub2の2パネルをBorderLayoutとして配置する．

(それぞれpanelNorth上のNORTH，CENTERに配置)

MainFrameはBorderLayout配置とし，上記の様なパネル配置とする．

panelNorthSub1への配置部品

教室予約システムラベル，ログインボタン，ログインIDラベル，ログインIDテキストフィールド

panelNorthSub2への配置部品

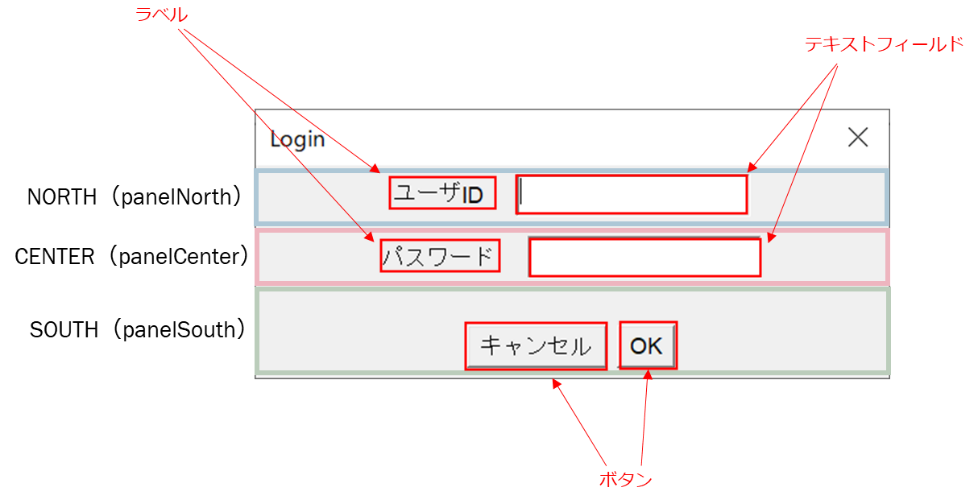
教室ラベル，教室選択コンボボックス，教室概要ボタン, 予約状況ボタン

panelCenterへの配置部品

テキストエリア（20行✕80文字）

panelSouthへの配置部品

* 1. ログインダイアログ

LoginDialogは，MainFrameと同様BorderLayout配置とし，下記の様なパネル配置及び部品配置とする．

panelNorthへの配置部品

　　ユーザーIDラベル、ユーザーIDテキストフィールド

　panelCenterへの配置部品

　　パスワードラベル、パスワードテキストフィールド

panelSouthへの配置部品

　キャンセルボタン、OKボタン

* 1. A screenshot of a computer

     Description automatically generated with medium confidenceReservation Dialog

青下線部 :ラベル

緑下線部 :テキストフィールド

赤下線部 :コンボボックス

紫下線部 :ボタン

panelNorth

panelCenter

panelSouth

Reservation DialogはMainFrameと同様BorderLayout配置とし，下記の様なパネル配置及び部品配置とする．

panelNorthへの配置部品

　教室ラベル、予約日ラベル、年ラベル、月ラベル、日ラベル、教室選択コンボボックス

　panelCenterへの配置部品

　　予約時間ラベル、(開始時間)時ラベル、分～ラベル、(終了時間)時ラベル、分ラベル、

　　(開始時間) 時選択コンボボックス、(開始時間)分選択コンボボックス、(終了時間)時時選択コン

　　ボボックス、(開始時間)分時選択コンボボックス

panelSouthへの配置部品

　空白ラベル、キャンセルボタン、予約実行ボタン

* 1. グラフィカル ユーザー インターフェイス

     自動的に生成された説明Reservation Status Dialog

panelNorth

panelCenter

panelSouth

青下線部 :ラベル

緑下線部 :テキストフィールド

赤下線部 :コンボボックス

紫下線部 :ボタン

Reservation Status DialogはMainFrameと同様BorderLayout配置とし，下記の様なパネル配置及 び部品配置とする．

panelNorthへの配置部品

　　教室ラベル、日付ラベル、年ラベル、月ラベル、日ラベル、教室選択コンボボックス、年テキス

　　トフィールド、月テキストフィールド、日テキストフィールド

　panelCenterへの配置部品

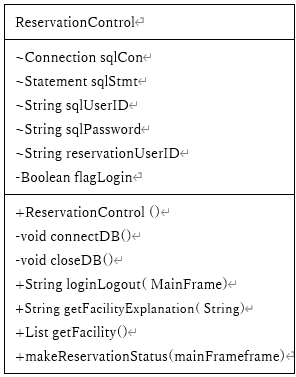
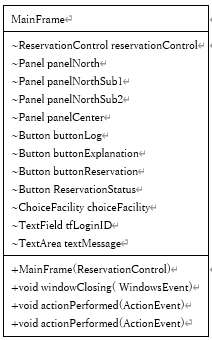
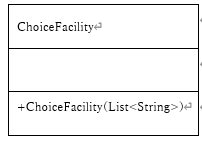
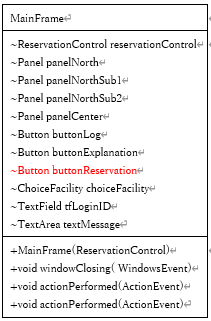
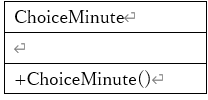
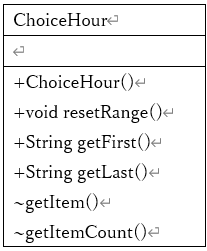
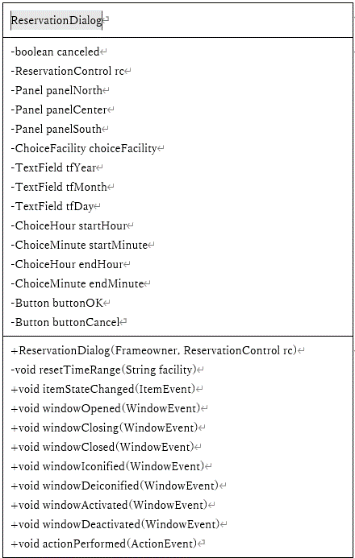
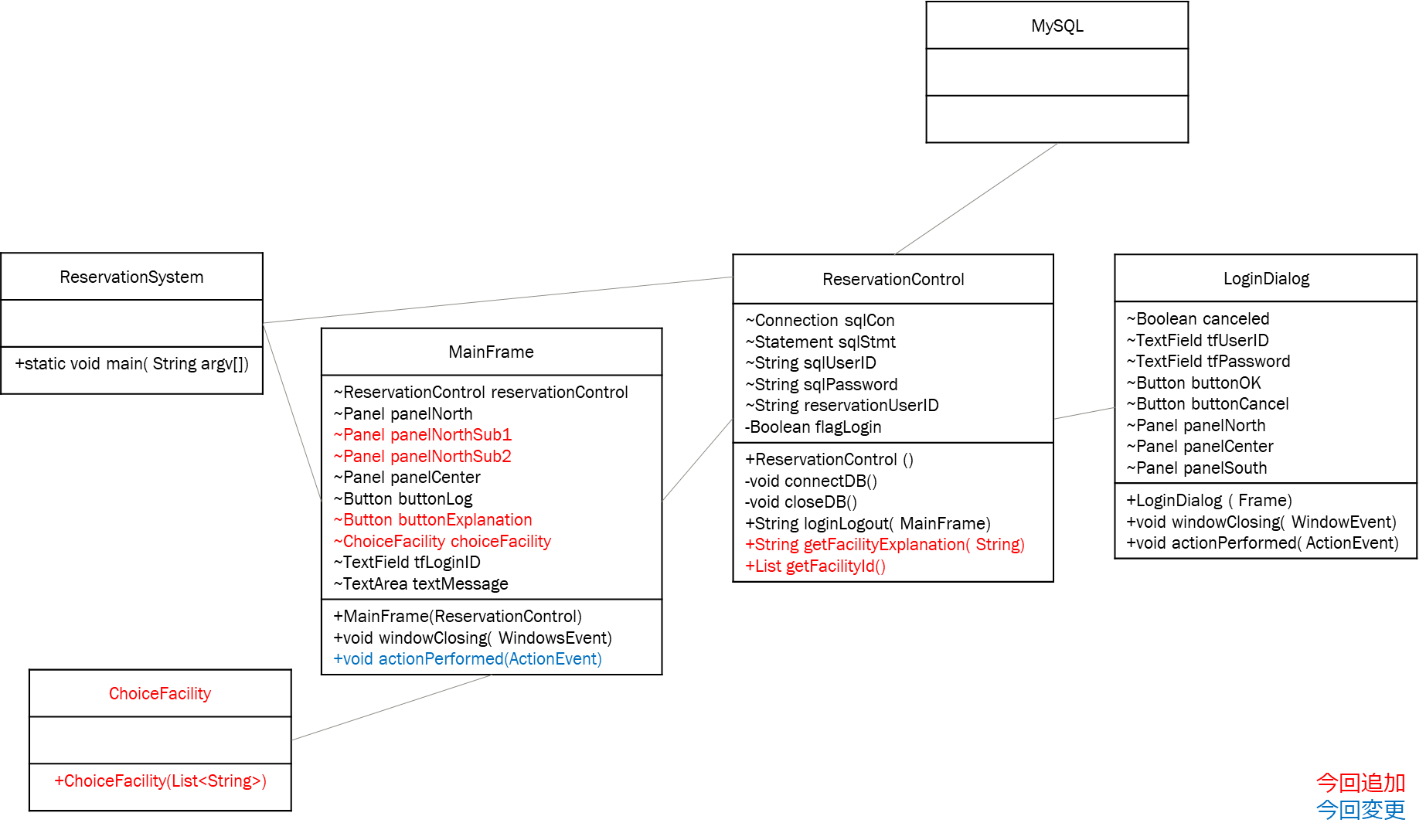
　　Reservation Status Dialogでは配置部品は無し

panelSouthへの配置部品

　空白ラベル、キャンセルボタン、OKボタン

1. クラス図及び各クラス内の変数及びメソッド

全体のクラス図を以下に示す．



* 1. ReservationSystem

|  |
| --- |
| ReservationSystem |
|  |
| +static void main( String argv[]) |

　ReservationSystemクラスは，本システムを起動するためのクラスで，MainFrameクラス及びReservationControlクラスを起動する役割を担う．それ以外の機能は持たない．

* + 1. mainメソッド

MainFrameクラス及びReservationControlクラスのインスタンスを生成し，MainFrameのWindowを表示，可視化を行う．

* 1. MainFrame

|  |
| --- |
| MainFrame |
| ~ReservationControl reservationControl  ~Panel panelNorth  ~Panel panelNorthSub1  ~Panel panelNorthSub2  ~Panel panelCenter  ~Button buttonLog  ~Button buttonExplanation  ~Button buttonReservation  ~Button ReservationStatus  ~ChoiceFacility choiceFacility  ~TextField tfLoginID  ~TextArea textMessage |
| +MainFrame(ReservationControl)  +void windowClosing( WindowsEvent)  +void actionPerformed(ActionEvent)  +void actionPerformed(ActionEvent) |

MainFrame classは，本システムの基本となるWindowを司るクラスである．MainFrameの各部品配置を行い，ユーザからのデータ入力及びボタン押下等のアクションに対して，該当するReservationControl classのメソッドを呼び出すことで，対応する処理を行う．

* + 1. MainFrameコンストラクタ

mainよりReservationControlクラスのSインスタンスを引数として受け取る．

コンストラクタ内で，MainFrameのWindow及び部品配置を行う．教室選択のコンボボックスは，Choiceクラスの派生クラスとして作成するChoiceFacilityクラスのインスタンスを生成し，実現する．ChoiceFacilityクラスのインスタンス生成時，ReservationControlのgetFacilityExplanationメソッドにより全てのfacility\_idを取得し，ChoiceFacilityのインスタンスに渡す．また，教室概要ボタン，ログインボタン及び終了ボタンを有効にするため，これらをActionListener及びWindowListenerに登録する．

* + 1. windowClosingメソッド

windowClosingメソッドは，終了ボタン押下時に呼び出され，本システムを終了させる．

* + 1. actionPerformedメソッド

actionPerformedメソッドは，ボタン押下時に呼出される．

ログインボタン押下時： reservationControlクラスのloginLogoutメソッドを呼び出す．

この時，戻り値を結果表示用テキストエリアに表示させる．

教室概要ボタン押下時：reservationControlインスタンスのgetFacilityExplanationメソッドを呼び出す．この時，戻り値を結果表示エリアに表示させる．

* 1. ReservationControl

|  |
| --- |
| ReservationControl |
| ~Connection sqlCon  ~Statement sqlStmt  ~String sqlUserID  ~String sqlPassword  ~String reservationUserID  -Boolean flagLogin |
| +ReservationControl ()  -void connectDB()  -void closeDB()  +String loginLogout( MainFrame)  +String getFacilityExplanation( String)  +List getFacility()  +makeReservationStatus(mainFrameframe) |

ReservationControl classは，本システムにてユーザが求める機能に対する処理を行うクラスである．MainFrameから，ボタン押下等の機能実行要求があった際に，本クラスのメソッドが呼出され，処理に必要な情報をMainFrameのTextField等の部品やMySQLとの通信を行うことで取得する．また処理した結果をMainFrameに表示する際は，本クラスのメソッドからの戻り値として文字列を返すことで実現する．その他，必要に応じて，MySQLに対してデータの書き込みも行う．

* + 1. sqlUserID

MySQLにログインするためのuserID格納変数．”puser”を入れておく．

* + 1. sqlPassword

MySQLにログインするためのuserIDに対応するパスワード格納変数．”1234”を入れておく．

* + 1. reservationUserID

本システムにログイン中のユーザIDを格納する変数．ログイン中のみ有効となる．

* + 1. flagLogin

本システムにユーザがログインしている状態かどうかを表す変数．ログイン中はtrueの値を取る．

* + 1. ReservationControlコンストラクタ

　flagLoginにfalseを設定し，非ログイン状態とする．

* + 1. connectDBメソッド

　MySQLのドライバをLoadし，その後MySQLとのconnectionを確立する．

connection確立後，Statement Objectを生成する．

上記いずれかの際にexceptionが発生した場合はStackTraceをコンソールに表示する．

* + 1. closeDBメソッド

Statement ObjectをCloseし，MySQLとのconnectionもcloseする．

上記いずれかの際にexceptionが発生した場合はStackTraceをコンソールに表示する．

* + 1. loginLogoutメソッド

MainFrameのログインボタン押下時にMainFrameにより呼出される．

ログイン状態時の処理：

　flagLoginにfalseを設定し，非ログイン状態にする．

また，ログインボタンの表記を「ログイン」に変更する．

ログインIDテキストエリアの表示を「未ログイン」に変更する．

　非ログイン状態時の処理：

ログインダイアログを生成し，制御をログインダイアログに移す．

ログイン操作キャンセルでダイアログが閉じられた時：

戻り値の文字列にNULLを設定し，処理を終了させる．

ログイン操作実行でダイアログが閉じられた時：

ログインダイアログで入力されたユーザIDとパスワードを取得する．

MySQLとのConnectionを確立する．

ログインダイアログに入力されたユーザIDと一致する情報をuserテーブルより取得．

MySQLに該当する情報がない時：

戻り値に「IDが違います」を設定する．

ログインダイアログに入力されたパスワードとuserテーブルのパスワードを比較する．

パスワードが一致：

ログイン状態（flagLogin）にtrueを設定する．

ログインボタンの表記を「ログアウト」に変更する．

ログインIDテキストフィールドにログインダイアログに入力したログインIDを設定する．

戻り値の文字列にNULLを設定する．

パスワードが不一致：

戻り値に「「IDまたはパスワードが違います．」を設定する．

MySQLとのConnectionをCloseする．

MySQLアクセス時にexceptionが発生した場合は，StackTraceをコンソールに表示する．

* + 1. getFacilityExplanationメソッド

　MainFrameの教室概要ボタン押下時にMainFrameにより呼び出され，引数として教室IDを受け取る．

本メソッドでは，MySQLに接続し，受け取った教室IDでfacilityテーブルを検索し，該当するデータのexplanation，open\_time，close\_timeより，表示用の教室概要データを作成し，戻り値として返す．

MySQLとの通信でexceptionが発生した場合は，コンソールにStackTraceを表示する．

* + 1. makeReservationstatusメソッド

MainFrameの新規予約ボタン押下時にMainFrameにより呼出される

ログイン状態に関わらず起動

予約状況メッセージを表示し、制御を予約ダイアログに移す.

予約操作キャンセルでダイアログが閉じられたとき:

戻り値の値に「””」空文字列を設定し、処理を終了させる.

西暦が4桁で入力されなかったとき：

戻り値の値に「日付の書式を修正して下さい」を格納

日付が正しく入力さなかった場合:

戻り値の値に「日付の値を修正して下さい」を格納.

日付が正しく入力された場合:

新規予約画面から、年月日を取得

月と日が一桁だった時、数字の前に０を付与しString型に合成

予約状況ダイアログを押したとき：

MySQLとのConnectionを確立する.

予約ダイアログに入力された情報をもとにreservationテーブルへ設定する

戻り値に「教室概要情報データ」を格納する.

MySQLとのConnectionを切断する.

予約ダイアログが入力されていない場合：

戻り値に「指定された教室、日付には予約が入っておりません」を格納

例外が発生したとき：

StackTraceを表示

MysqlとのConnectionを切断

* 1. LoginDialog

|  |
| --- |
| LoginDialog |
| ~Boolean canceled  ~TextField tfUserID  ~TextField tfPassword  ~Button buttonOK  ~Button buttonCancel  ~Panel panelNorth  ~Panel panelCenter  ~Panel panelSouth |
| +LoginDialog ( Frame)  +void windowClosing( WindowEvent)  +void actionPerformed( ActionEvent) |

　LoginDialog classは，ユーザがログインする際にユーザIDやパスワードを入力するためのダイアログに於いて，Viewの役割を果たす．ユーザID及びパスワード入力後， 本ダイアログが閉じられた要因に従い，変数canceledにログイン処理を行うかキャンセルするかを設定する．

* + 1. canceled

　この変数はログインダイアログを閉じる際に，ユーザがログイン処理を行うかログイン処理をキャンセルするかを決めるためのものである．ログインダイアログが終了ボタンもしくはキャンセルボタンで閉じられた場合はtrueが設定され，以降ログイン処理はキャンセル扱いとなる．また，OKボタンにて閉じられた場合はfalseが設定され，ログイン認証動作に移行する．

* + 1. LoginDialogコンストラクタ

　本クラスは，Dialogクラスの派生クラスで，ActionListener及びWindowListenerを利用する．引数でMainFrameのインスタンスを取得し，親クラスであるDialogのコンストラクタを呼び出し，本ダイアログの所有者，タイトル，モーダル指定を行う．

（所有者：MainFrame，タイトル：Login，モーダル指定：有効（true））

　ダイアログの各種部品配置及びActionListenerやWindowListenerへのボタン登録を行う．（ActionListenerへは，キャンセルボタン及びOKボタンが設定される．）

* + 1. windowClosingメソッド

　windowClosingメソッドは，終了ボタン押下時に呼出され，ログインダイアログを非表示に設定，canceled変数にtrueを設定，ダイアログを削除する．

* + 1. actionPerformedメソッド

　actionPerformedメソッドは，キャンセルボタンもしくはOKボタン押下時に呼出される．

canceled変数に，キャンセルボタン押下時はtrue，OKボタン押下時はfalseが設定され，その後ログインダイアログを非表示にし，ダイアログを削除する．

* 1. ChoiceFacility

　ChoiceFacility classは，Choiceクラスを継承した派生クラスで，教室を選択するコンボボックスの機能を提供する．本クラス独自のメソッドは持たず，選択された教室IDはChoiceクラスのgetSelectedItemメソッドにて取得することが出来る．[システム構造設計書\_v2.0.docx](https://nkzacjp-my.sharepoint.com/:w:/g/personal/tk210376_tks_iput_ac_jp/ETjI-yZGkuZMoBf3AnI-No8BK75t72fqzImGD4t959b9Sg?e=MdNjSH)

* + 1. ChoiceFacilityコンストラクタ

呼出元から受け取ったリストにあるfacility\_idを全てchoiceのアイテムに加える．

|  |
| --- |
| ChoiceFacility |
|  |
| +ChoiceFacility(List<String>) |

* 1. ReservationDialog

|  |
| --- |
| ReservationDialog |
| -boolean canceled  -ReservationControl rc  -Panel panelNorth  -Panel panelCenter  -Panel panelSouth  -ChoiceFacility choiceFacility  -TextField tfYear  -TextField tfMonth  -TextField tfDay  -ChoiceHour startHour  -ChoiceMinute startMinute  -ChoiceHour endHour  -ChoiceMinute endMinute  -Button buttonOK  -Button buttonCancel |
| +ReservationDialog(Frameowner, ReservationControl rc)  -void resetTimeRange(String facility)  +void itemStateChanged(ItemEvent)  +void windowOpened(WindowEvent)  +void windowClosing(WindowEvent)  +void windowClosed(WindowEvent)  +void windowIconified(WindowEvent)  +void windowDeiconified(WindowEvent)  +void windowActivated(WindowEvent)  +void windowDeactivated(WindowEvent)  +void actionPerformed(ActionEvent) |

ReservationDialog classは新規予約を作成するためのダイアログを表示される。教室選択、予約日、予約時間（開始・終了）をユーザーに入力させるGUIコンポーネントを所持している。また、予約を実行するためのボタンと、ダイアログをキャンセルするためのキャンセルボタンの機能も提供する。

* + 1. ReservationControl コンストラクタ

親クラス（Dialog）のコンストラクタを呼び出し、ダイアログボックスをモーダルで初期化し、ReservationControl のインスタンスを rc フィールドに保存。GUIの各種部品生成配置を行う。新規予約のキャンセルステータスを初期化します。選択ボックスとボタンはユーザーの操作に対する応答を可能させる。初期化時にユーザーが選択している教室の利用可能時間を設定される。ダイアログのサイズを設定し、サイズ変更を無効にする。

* + 1. **resetTimeRange(String facility)　メソッド**

指定された教室の利用可能時間を取得し、それを基に予約開始時間と終了時間の選択範囲を設定可能。

* + 1. **itemStateChanged(ItemEvent)　メソッド**

GUIの選択肢が変更されたときの挙動を定義している。ユーザーが操作を行い、他の選択肢の範囲や値を適切に調整する機能。

choiceFacility.addItemListener(this); 施設選択ボックスが変更されたとき、選択された教室の利用時間に合わせ、開始時間と終了時間の選択範囲を再設定する。設定後の時間が選択範囲外になった場合、範囲に収まるように修正する。

startHour.addItemListener(this); 開始時間選択ボックスが変更されたとき、終了時間の選択範囲を開始時間以降に再設定する。設定後の終了時間が選択範囲外になった場合、開始時間と同じに設定する。

endHour.addItemListener(this); 終了時間選択ボックスが変更されたとき、開始時間の選択範囲を終了時間以降に再設定する。設定後の終了時間が選択範囲外になった場合、終了時間と同じに設定する。

* + 1. **windowClosing(WindowEvent) メソッド**

ウィンドウが閉じられるとき非表示になり、ウィンドウとそのすべての子コンポーネントがグラフィックスリソースを解放される。

* + 1. actionPerformed(ActionEvent) メソッド

GUIのOKかキャンセルボタンがクリックされた場合の挙動を定義している。

キャンセルボタンが押された場合、予約作成をキャンセル。入力された情報をクリアし、ダイアログを閉じる。

OKボタンが押された場合、予約情報の確定処理を行う。選択された教室、予約日、開始時間、終了時間を取得し、新しい予約をシステムに登録する。

* + 1. 使用されてないメソッド

+void windowClosed(WindowEvent)

+void windowIconified(WindowEvent)

+void windowDeiconified(WindowEvent)

+void windowActivated(WindowEvent)

+void windowDeactivated(WindowEvent)

+void windowOpened(WindowEvent)

* 1. ChoiceHour Class

|  |
| --- |
| ChoiceHour |
|  |
| +ChoiceHour()  +void resetRange()  +String getFirst()  +String getLast()  ~getItem()  ~getItemCount() |

ChoiceHour Classはchoiceクラスの派生クラスで、予約時を指定するChoiceに9～21を登録する．

登録した中から選択されたものを文字列として選択ボックスに追加する．

* + 1. ChoiceHourコンストラクタ

選択できる時刻の範囲を9～21とする

* + 1. resetRangeメソッド

変数Startと変数endに登録されている項目を全削除後、変数startをLoop変数として＋１しながらendと同値になるまで項目に文字列として登録する．この時、変数Startはstring型変数hに格納され、その値が一桁である時、数字の前に0を付加する．

* + 1. getFirstメソッド

登録されている時間の中で一番早い時間、つまり0番目のデータを返す

* + 1. getLastメソッド

登録されている時間の中で一番遅い時間を返す

* + 1. getItemメソッド

引数で指定した項番のデータを 返す．この場合は先頭データを返す．

* + 1. getItemCountメソッド

　　 項⽬の個数を返す

* 1. ChoiseMinute Class

|  |
| --- |
| ChoiceMinute |
|  |
| +ChoiceMinute() |

* + 1. ChoiceMinuteコンストラクタ

　Choiceクラスの派生クラスで、「00」「15」「30」「45」という値を項目に追加する

* 1. ReservationStatusDialog Class

時間入力が入っていないこと以外はReservationDialogと同じため一部割愛

|  |
| --- |
| ReservationStatusDialog |
|  |
| -void itemstateChanged(ItemEvent e) |

* + 1. itemstateChange

自動生成されたメソッド・スタブ

1. シーケンス図

以下にシーケンスのパターンを列挙し，次ページよりシーケンス図を示す．

4.1起動

4.2 ログイン機能

4.2.1 ログインダイアログ表示

4.2.2 ID,パスワード入力～認証完了

4.2.3 ID，パスワード入力～ユーザID未登録

4.2.4 ID，パスワード入力～パスワード不一致

4.2.5 認証中止

4.3 教室概要表示

4.4 予約機能

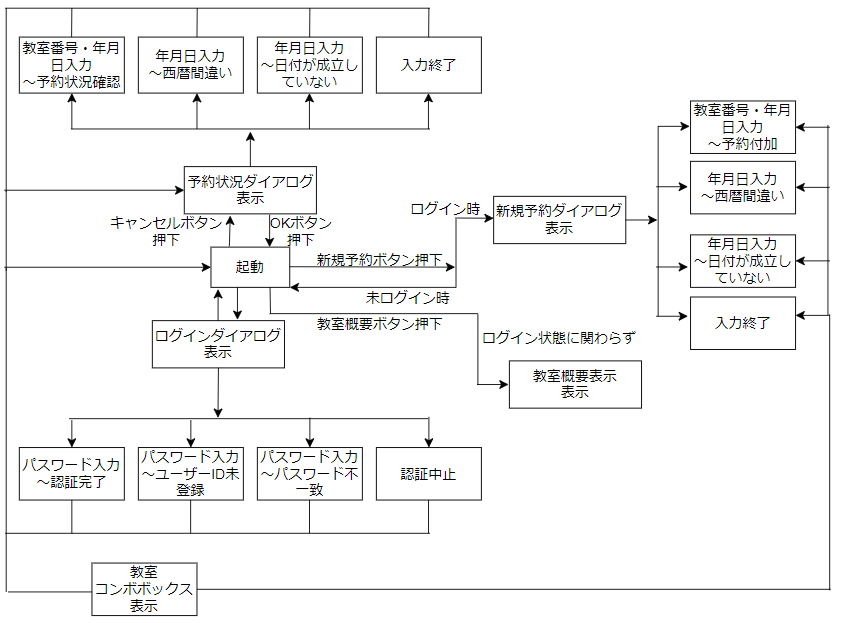
4.5 教室予約状況確認

各シーケンス間の繋がりを以下に示す．

ログイン状態

未ログイン状態

予約状況ボタン押下



4.1

起動

ログインボタン押下

教室番号/年月日入力

ダイアログ表示

キャンセルボタン押下

4.2.1

ログイン

ダイアログ表示

4.2.5

認証中止

4.2.2

パスワード入力

～認証完了

4.2.4

パスワード入力

～パスワード不一致

4.2.3

パスワード入力

～ユーザID未登録

教室概要ボタン押下

ログイン状態

教室概要表示

教室

コンボボックス表示

未ログイン状態

教室概要表

コンボボックスボタン押下

4.1起動

MainFrame

ChoiceFacility

LoginDialog

ReservationControl

MySQL

ReservationSystem

reservationControlインスタンス生成

*flagLoginを非ログインに設定*

mainFrameインスタンス生成

全てのfacilityデータ取得

connectDB(DBに接続)

closeDB(DBから切断)

sqlStmt.executeQuery

getFacilityId()

*facilityテーブルの全てのデータ取得*

ChoiceFacilityインスタンス生成(全facility\_id)

全てのfacility\_idデータ

*ボタン，Windowの入力リスナを登録する．*

*ボタン・テキストフィールド等を組み立てる．*

setVisible(Window可視化)

setBounds(サイズ･表示位置設定)

*教室番号をChoiceの項目に追加*

グラフィカル ユーザー インターフェイス, テキスト

中程度の精度で自動的に生成された説明

4.2ログイン機能

4.2.1ログインダイアログ表示

本シーケンスは，MainFrame表示中かつ非ログイン状態のものである．（ex.4.1 起動からの継続）

ReservationSystem

MySQL

ReservationControl

LoginDialog

ChoiceFacility

MainFrame

ldインスタンス生成

*flagLoginが非ログインの時*

loginLogout(this)

ログインボタン押下

setResizable(Windowリサイズ不可)

setBounds(サイズ・表示位置設定)

グラフィカル ユーザー インターフェイス, テキスト

中程度の精度で自動的に生成された説明

setVisible(Window可視化)

setModalityType (他Windowの操作禁止)

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

4.2.2ID，パスワード入力～認証完了

本シーケンスは，ログインダイアログ表示中に正しいパスワードを入力し，OKボタンを押下した時のものである．（ex.4.2 ログインダイアログ表示からの継続）

ReservationSystem

MainFrame

LoginDialog

ReservationControl

MySQL

ChoiceFacility

*canceledにfalseを設定*

*ログインWindowを非表示にし，消去*

ユーザがIDとパスワードを入力し，「OK」ボタン押下

入力ユーザID

tfUserID.getText(入力ユーザID取得)

connectDB(DBに接続)

入力パスワード

tfPassword.getText(入力パスワード取得)

*取得したタプルのパスワードと*

*入力パスワードが一致したら，*

*flagLoginをログイン中に設定*

入力ユーザIDと合致するタプル

と合致するタプルを取得

SELECT文で入力ユーザID

sqlStmt.executeQuery

closeDB(DBから切断)

tfLoginID.setText

*ログインIDテキストフィールドにユーザIDを表示*

*ログインボタンのラベルを「ログアウト」に設定*

buttonLog.setLabel

4.2.3 ID，パスワード入力～ユーザID未登録

「IDまたはパスワードが違います．」

本シーケンスは，ログインダイアログ表示中に未登録のユーザIDを入力し，OKボタンを押下した時のものである．（ex.4.2 ログインダイアログ表示からの継続）

ReservationSystem

MainFrame

LoginDialog

ReservationControl

MySQL

ChoiceFacility

*canceledにfalseを設定*

*ログインWindowを非表示にし，消去*

ユーザがIDとパスワードを入力し，「OK」ボタン押下

入力ユーザID

tfUserID.getText(入力ユーザID取得)

connectDB(DBに接続)

入力パスワード

tfPassword.getText(入力パスワード取得)

*SELECT文で戻りデータが*

*無いということは，指定した*

*IDが存在しないと判断．*

データ無し

と合致するタプルを取得

SELECT文で入力ユーザID

sqlStmt.executeQuery

closeDB(DBから切断)

「IDが違います．」

4.2.4 ID，パスワード入力～パスワード不一致

本シーケンスは，ログインダイアログ表示中に登録済みユーザID及び誤ったパスワードを入力し，OKボタンを押下した時のものである．（ex.4.2 ログインダイアログ表示からの継続）

ReservationSystem

MainFrame

LoginDialog

ReservationControl

MySQL

ChoiceFacility

*canceledにfalseを設定*

*ログインWindowを非表示にし，消去*

ユーザがIDとパスワードを入力し，「OK」ボタン押下

入力ユーザID

tfUserID.getText(入力ユーザID取得)

connectDB(DBに接続)

入力パスワード

tfPassword.getText(入力パスワード取得)

*取得したタプル内のパスワード*

*が入力されたパスワードと*

*等しくない．*

入力ユーザIDと合致するタプル

と合致するタプルを取得

SELECT文で入力ユーザID

sqlStmt.executeQuery

「IDまたはパスワードが違います．」

closeDB(DBから切断)

4.2.5認証中止

本シーケンスは，ログインダイアログ表示中にキャンセルボタンを押下した時のものである．（ex.4.2 ログインダイアログ表示からの継続）

ReservationSystem

MainFrame

LoginDialog

ReservationControl

MySQL

ChoiceFacility

*canceledにfalseを設定*

「キャンセル」ボタン押下

*ログインWindowを非表示にし，消去*

*MainFrameのテキストフィールドに*

*戻り値を表示．*

「」（文言なし）

4.3教室概要表示

本シーケンスは，MainFrameのみ表示中の状態で，コンボボックスより教室選択を行い，教室概要ボタンを押下した時のものである．（ex.4.1 起動後からの継続）

ReservationSystem

MainFrame

LoginDialog

ReservationControl

MySQL

ChoiceFacility

教室概要ボタン押下

教室選択内容変更

選択された教室ID

getSelectedItem()

getFacilityExplanation(教室ID)

closeDB(DBから切断)

explanation,open\_time,close\_timeデータ

*IDと合致するタプルを取得*

sqlStmt.executeQuery

入力ユーザIDと合致するタプル

connectDB(DBに接続)

*SELECT文で選択された教室*

選択された教室IDの

*テキストエリアに教室概要*

*データを表示*

4.4新規予約（簡易版）

4.4.1 未ログイン状態からの新規予約ボタン押下

ダイアグラム が含まれている画像

自動的に生成された説明

ReservationControl

MySQL

closeDB(DBから切断)

sqlStmt.executeQuery

connectDB(DBに接続)

ReservationSystem

MainFrame

ChoiceFacility

flagloginを非ログインに設定

ReservationControl

インスタンス作成

MainFrame

インスタンス作成

選択された教室ID

getFacilityId

*facilityの全データ取得*

setVisible (Window)

全てのfacilityデータ取得

全てのfacilityIdデータ

全てのFacility

データ

choiseFacilityインスタンス作成

全てのFacility

データ

setBounds (サイズ・表示位置設定)

ボタン、テキストフィールド等を組み立てる

ボタン、Windowリスナーを

登録する

新規予約ボタン押下

4.4.2 ログイン後新規予約ボタン押下〜予約実⾏まで

ダイアグラム

中程度の精度で自動的に生成された説明

4.4.3 選択教室変更

グラフィカル ユーザー インターフェイス, ダイアグラム, アプリケーション

自動的に生成された説明

4.4.4開始「時」変更～開始「分」変更～予約実行まで

ダイアグラム

中程度の精度で自動的に生成された説明

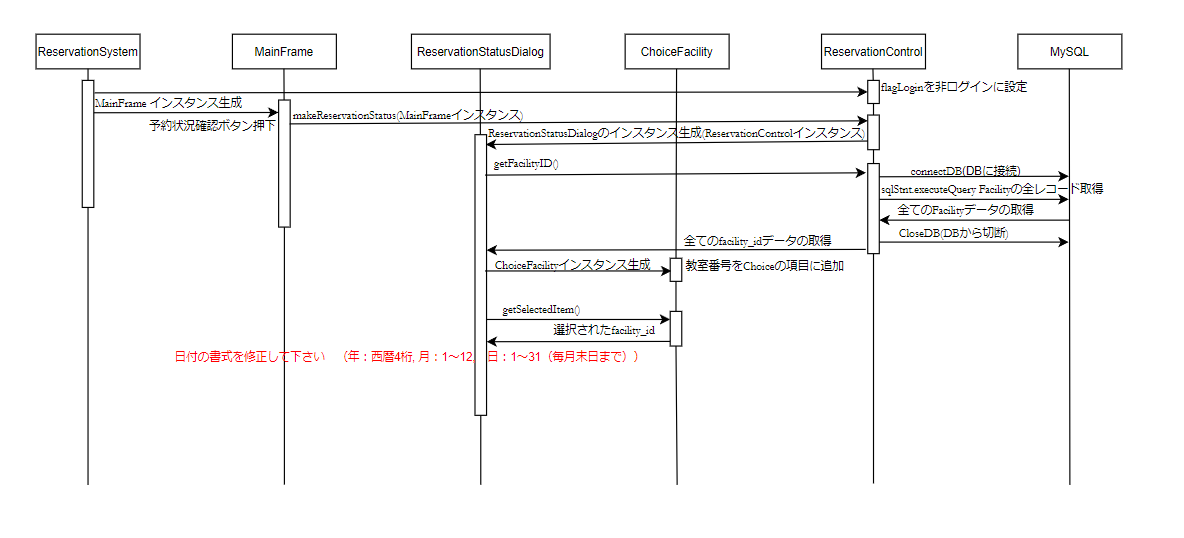
4.4.5　新規予約ボタン押下～入力データチェック～予約実行まで

ダイアグラム

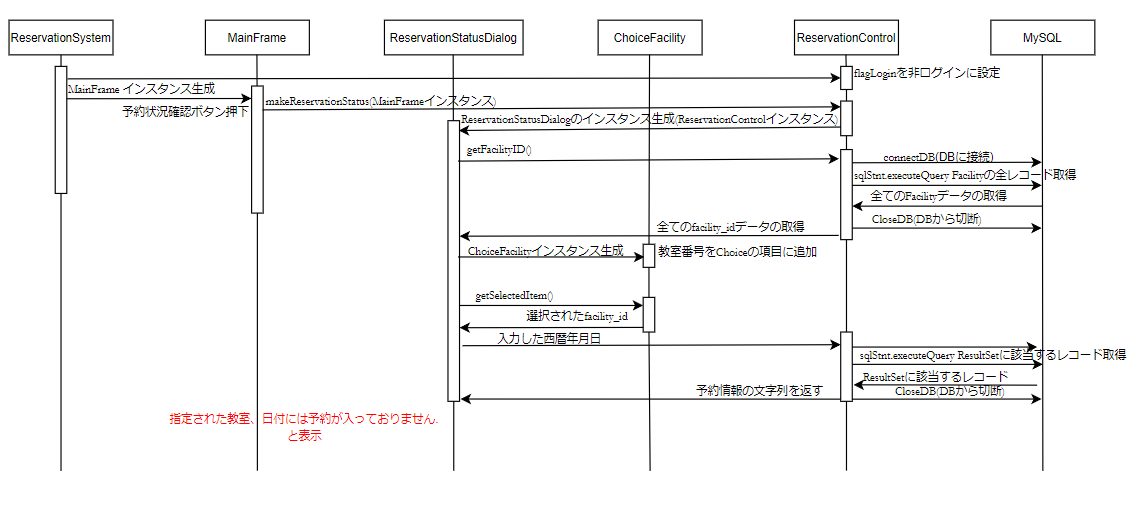
自動的に生成された説明

4.5 教室予約状況確認

　　4.5.1 日付情報入力～日付情報不一致



　　4.5.2 予約情報なし



　　4.5.3 予約状況ボタン押下～予約情報表示

