

【概要】

勤怠管理システムWebアプリ

【特徴】

- ・従業員自身の希望シフトを容易に申請登録できる
- ・シフト管理者が従業員のシフトを組みやすく、作成時間削減が実現できる
- ・全員のシフトを確認できる
- ・チャットシステム導入により、シフト変更希望などを連絡できる
- ・調子アイコンで従業員の気分を表現できる

【対象者】

シフト管理者、バイト従業員

【機能概略】

- ・ログイン画面

ユーザーID、パスワードを入力してログイン

ユーザーIDは個々に管理、パスワードは共通

ログイン時点で従業員ページと管理者ページに分かれる

- ・従業員用シフト画面トップページ

従業員一覧表示、シフトをカレンダー表示

自分のところだけ編集可能。調子変更可能。

従業員用シフト登録画面へリンクするボタンがある

- ・従業員シフト登録画面

管理者がシフト作成開始連絡後、希望シフトを登録できる

自分のシフト、従業員のシフトを確認できる

- ・管理者用シフト画面トップページ

従業員一覧表示、シフトをカレンダー表示。

従業員全員のシフトを個人ページで編集可能

リンクボタン→従業員新規登録画面、シフト項目登録画面、ランク登録、

シフト作成開始画面、管理者ID・パスワード変更画面

- ・従業員情報個人ページ（管理者のみ）

従業員の基本情報確認、編集、シフト修正ボタンがある
(シフト修正ボタンより希望シフト確認、登録ができる)

- ・従業員新規登録画面（管理者のみ）

新人雇用の際に従業員を新規登録できる
ID、名前、性別、基本シフト、ランク
退職後のデータ削除ボタン

- ・シフト作成開始画面（管理者のみ）

シフト作成開始/終了ボタン
従業員がシフトを登録できる期間を設定できる
希望シフト未提出者数が反映される→提出を促す発信ボタンがある
「シフト公開」ボタンで従業員に一斉にシフト表が反映される

- ・シフト項目登録画面編集ページ（管理者のみ）

シフト種類、時間帯を登録できる

- ・シフト登録画面

1か月ごとにそれぞれのシフトを登録できる

- ・社内専用チャット（従業員画面、管理者画面共通）

シフト変更希望などのメッセージを送れる

- ・お知らせ欄

シフト未提出者へ催促メッセージを反映できる

- ・その他

PCブラウザ用、スマートフォン表示用に自動で画面レイアウトが変更される
BGMが流れる
ログイン時間によって、あいさつ文が変わる

【チームメンバー】

梅本、古田、小寺、政島、志水、濱田

【開発環境】

eclipse Tomcat9_Java11
VS CODE

管理者ページ

ログイン画面

社名

ユーザーID:

パスワード:

ログイン

ログイン成功

MAIN画面 管理者トップページ

1月5日(金) ○○○さんようこそ!

2024年1月シフト 調整中 確定

社内専用チャット

名前

	1/1	1/2	1/3	1/4	1/5	1/6	1/7
調子	(日)	(月)	(火)	(水)	(木)	(金)	(土)
○○○○	日勤	日勤	日勤	日勤	日勤	日勤	日勤
△△△	早朝	早朝	早朝	早朝	早朝	早朝	早朝
▽▽▽	夜勤	夜勤	夜勤	夜勤	夜勤	夜勤	夜勤
◇◇◇	早朝	10	10	10	10	10	5
△△△	日勤	5	5	5	5	5	10
▽▽▽	夜勤	10	10	10	10	10	10

従業員新規登録 シフト項目登録

シフト作成開始 パスワード変更 管理人ID変更

株式会社

チーム名どうする

ログイン画面

社名

ユーザーID:

パスワード:

ログイン

失敗の場合

ログイン画面

社名

ユーザー名またはパスワードが違います

ユーザーID:

パスワード:

ログイン

従業員新規登録ページ

管理者シフト編集ページ

1月5日(金)

ID: 名前:

性別:

ランク:

NGリスト

調子:

登録 閉じる

本当に削除してもよろしいですか?

いいえ はい

データ削除

株式会社

チーム名どうする

従業員情報

1月5日(金)

ID: 01 ○○○○ あいうえお 編集

性別: 男

ランク: 編集

NGリスト: 編集

調子: シフト修正

閉じる

株式会社

チーム名どうする

シフト作成開始画面

休日設定

選択中の年月 2024年1月

日	月	火	水	木	金	土	日
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	

設定可能期間 12/20 00:00 ~ 12/20 00:00

通知開始

戻る

提出済 未提出 対象者

20 5 25

未提出者選及

シフト公開

株式会社

チーム名どうする

パスワード変更

現在のパスワード:

新しいパスワード:

新しいパスワード(確認):

変更

閉じる

株式会社

チーム名どうする

シフト登録画面

ID: 01 ○○○○

2024年1月シフト 変更 戻る

日	月	火	水	木	金	土
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

決定

株式会社

チーム名どうする

シフト項目登録画面編集ページ

シフト名登録: 日勤

時間帯: 11:00 ~ 18:00

備考:

閉じる 登録 修正

株式会社

チーム名どうする

管理人ID変更

現在のID:

新しいID:

パスワード:

変更

閉じる

株式会社

チーム名どうする



シフト管理システム

ID :

PW :

Login

失敗

ログインに失敗しました！

シフト管理システム

ID :

PW :

Login

成功

MAIN画面 トップページ (PC)

1月5日(金) 2024年1月シフト シフト登録 ログアウト

〇〇さんようこそ

名前 1/1 1/2 1/3 1/4 1/5 1/6 1/7

名前	調子(日)	1/1	1/2	1/3	1/4	1/5	1/6	1/7
〇〇	調子	日勤	早朝	夜勤	夜勤	早朝	早朝	日勤
●●	調子	早朝	日勤	夜勤	夜勤	早朝	早朝	日勤
▽▽	調子	夜勤	夜勤	夜勤	夜勤	早朝	早朝	日勤
◆◆	調子	夜勤	夜勤	夜勤	夜勤	早朝	早朝	日勤

社内専用チャット

SHANINE

コメント

中東田徳貴 : 01/29 10:11:36
こんにちは！

01/29 10:11:39
koko

中東田徳貴 : 01/29 10:12:41
こんにちは！

中東田徳貴 : 01/29 10:12:51
調子がわるい

MAIN画面 トップページ (スマホ→600px以下?)

1月5日(金) 〇〇〇〇さん ようこそ！

2024年1月シフト シフト登録 シフト修正依頼 ログアウト

名前 1/1 1/2 1/3 1/4 1/5 1/6 1/7

名前	調子(日)	1/1	1/2	1/3	1/4	1/5	1/6	1/7
〇〇	調子	日勤	早朝	夜勤	夜勤	早朝	早朝	日勤
●●	調子	早朝	日勤	夜勤	夜勤	早朝	早朝	日勤
▽▽	調子	夜勤	夜勤	夜勤	夜勤	早朝	早朝	日勤
◆◆	調子	夜勤	夜勤	夜勤	夜勤	早朝	早朝	日勤

社内専用チャット

SHANINE

コメント

中東田徳貴 : 01/29 10:11:36
こんにちは！

01/29 10:11:39
koko

中東田徳貴 : 01/29 10:12:41
こんにちは！

中東田徳貴 : 01/29 10:12:51
調子がわるい

ログアウトしました

シフト管理システム

ID :

PW :

Login

シフト登録・修正 画面

シフト登録画面

ID : 01 〇〇〇〇 2024年1月シフト 変更 戻る

日	月	火	水	木	金	土
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

決定

提出

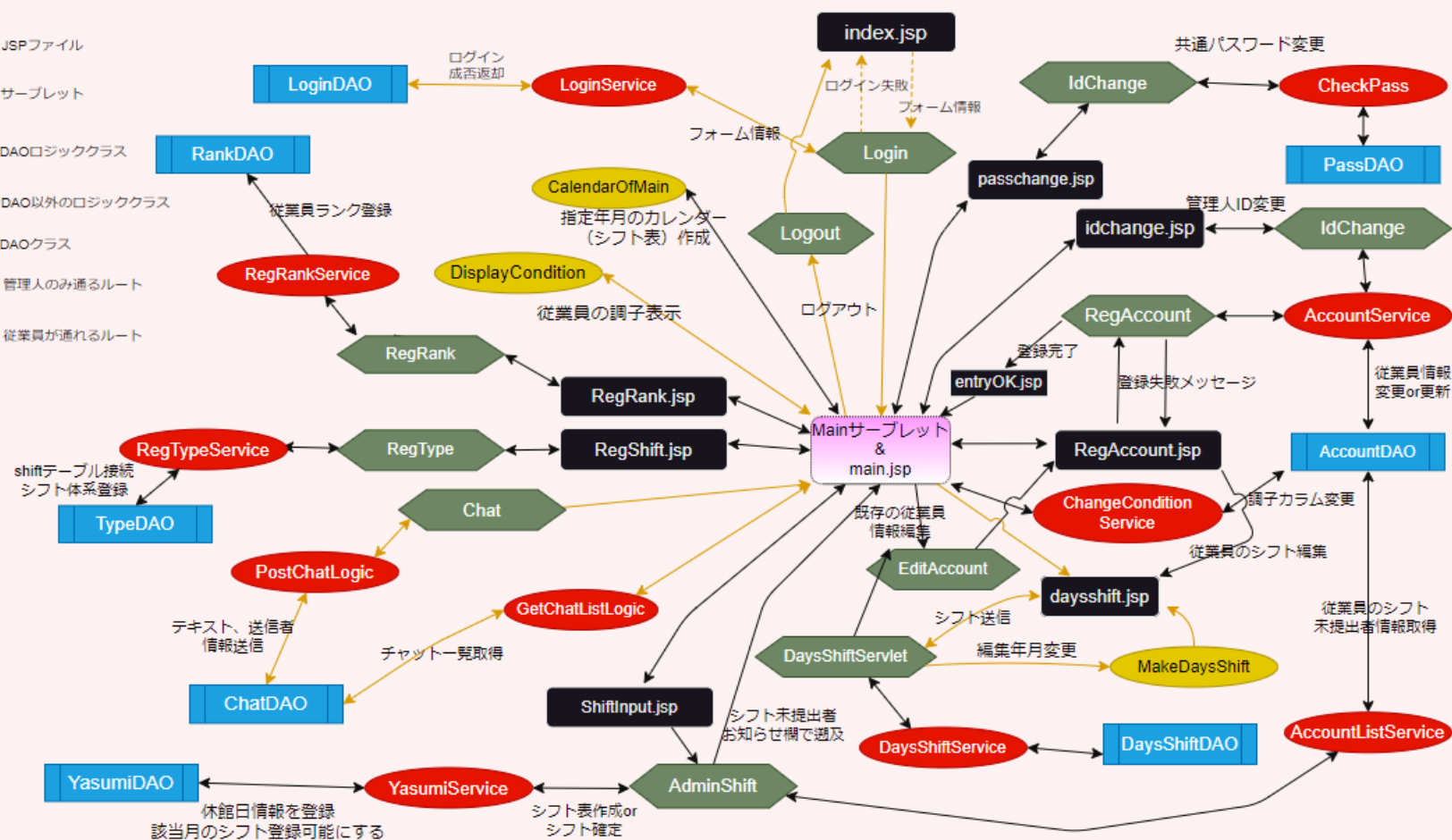
株式会社
チーム名どうする

シフト登録画面 (スマホ→600px)

2024年1月シフト MAINへ

早朝 日勤 夜勤 消す 提出

日	月	火	水	木	金	土
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			



ID	1				
クラス名	AccountDAO				
メソッド名	insert	ChangeCondition	update	updateAdmin	findByld
引数の型	Account	Account	Account	String	String
戻り値型	boolean	boolean	boolean	boolean	account
内容	従業員登録フォームで入力した情報を accountテーブルに登録 問題なく登録されればtrue	accountテーブルと接続 ログイン中の従業員のIDと合致する データのcondiカラムを変更、問題なければtrue	accountテーブルと接続 新規登録する従業員の情報を送信 insertが完了すればtrue	accountテーブル内の管理人のldを変更 問題なければtrue	accountテーブル内に引数のidと 合致するアカウントがあればそ れを返す
ID					
メソッド名	getAll	getAllAccountWithYM	getCountIfExistList	getNameIfNotExistList	
引数の型		String, String	String, String	String, String	
戻り値型	List<Account>	List<MonthliShift>	int	List<String>	
内容	accountテーブル内に登録されているアカウントの 情報をすべて取得	引数で指定した年、月の アカウントの名前、月単位のシフト内容を リストにして一覧返却	引数で指定した年、月のシフトを 提出しているアカウントの数をカウン トして返却	引数で指定した年、月のシフトを 提出していないアカウントの名前を リストにして取得	

ID	2	
クラス名	LoginDAO	
メソッド名	findByID	
引数の型	LoginUser	
戻り値型	Account	
内容	Accountテーブルと接続 ログインフォームの内容とデータベースで合致する アカウントがあればログイン成功、モデル返却	

ID	3	
クラス名	RankDAO	
メソッド名	insert	findAll
引数の型	Shift	
戻り値型	boolean	List<Rank>
内容	gradeテーブルと接続	gradeテーブルと接続

ID	4	
クラス名	TypeDAO	
メソッド名	insert	getAll
引数の型	ShiftType	なし
戻り値型	boolean	List<ShiftType>
内容	shiftテーブルと接続	shiftテーブルと接続

ID	5	
クラス名	PassDAO	
メソッド名	verificate	update
引数の型	String	String
戻り値型	boolean	boolean
内容	passテーブルと接続 ログイン時に入寮されたパスワードがDBのパスと 一致しているかを照合して成否を返却	共通パスワード変更

ID	6			
クラス名	DaysShiftDAO			
メソッド名	insert	countType	findByYM	update
引数の型	DaysShift	ShiftType, String, String	String, String, String	DaysShiftDAO
戻り値型	boolean	List<Integer>	boolean	boolean
内容	dayshiftテーブルと接続 従業員の該当月のシフト情報を登録 問題なければtrue	指定した年月で、各シフトタイプが日ごとに何人 登録されているかをカウントしてリストとして返 却する	アカウントID,年月をもとに シフト表がテーブルに存在しているか をBooleanで返す	すでに存在している従業員の該当月のシ フト内容を更新、問題なければtrue

ID	7			
クラス名	YasumiDAO			
メソッド名	insert	updateConfirmed	getWithYM	updateYasumi
引数の型	Yasumi	String, String	String, String	Yasumi
戻り値型	boolean	boolean	Yasumi	boolean
内容	yasumiテーブルと接続 該当月の休日情報を登録 問題なければtrue	yasumiテーブルと接続 該当月のシフトを確定。従業員は変更不可にする 問題なければtrue	引数で指定した年月の休日情報を取得	すでに存在するシフト表の情報を 更新する

ID	8	
クラス名	ChatDAO	
メソッド名	create	findAll
引数の型	Chat	
戻り値型	boolean	List<Chat>
内容	chatテーブルと接続 送信されてきたチャット情報をDBに送信 問題なければtrue	chatテーブルと接続 格納されているチャット情報一覧を取得

ID	1
クラス名	ChangeConditonService
メソッド名	execute
引数の型	Account
戻り値型	boolean
接続するDAO	AccountDAO
内容	dao.changeConditionを実行 ログインアカウントの調子変更

ID	2
クラス名	LoginService
メソッド名	execute
引数の型	LoginUser
戻り値型	Account
接続するDAO	LoginDAO
内容	findByld(loginUser)を実行 ログイン情報と合致する アカウントを返す

ID	3				
クラス名	AccountService				
メソッド名	checkContext	insert	update	findByld	updateAdmin
引数の型	Account	Account	Account	String id	String
戻り値型	boolean	boolean	boolean	AccountDAO	boolean
接続するDAO	なし	AccountDAO	AccountDAO	AccountDAO	AccountDAO
内容	Accountインスタンス内の name.idがnullでないかをチェック 問題なければtrue	dao.insert(account)を実行 新規従業員を追加 問題なければtrue	従業員情報を更新する 問題なければtrue	引数で受け取った従業員idをもとに データベースを検索。見つければそ の情報をAccountで返却	管理人のIdを変更する

ID	4	
クラス名	RankService	
メソッド名	insert	findAll
引数の型	RankService	
戻り値型	boolean	List<Rank>
接続するDAO	RankDAO	RankDAO
内容	dao.insert(grade)を実行 新規ランクを登録 問題なければtrue	登録されているランクをすべて取 得

ID	5	
クラス名	TypeService	
メソッド名	insert	findAll
引数の型	ShiftType	
戻り値型	boolean	List<ShiftType>
接続するDAO	TypeDAO	TypeDAO
内容	dao.insert(shift)を実行 新規シフト体系を登録 問題なければtrue	登録されているシフト体系をすべ て取得

ID	6			
クラス名	AccountListService			
メソッド名	getAll	getAllAccountWithYM	getCountIfExistList	getNamelIfNotExistList
引数の型	無し	String y, String m	String y, String m	String y, String m
戻り値型	List<Account>	List<MonthlyShift>	int	List<String>
接続するDAO	AccountDAO	AccountDAO	AccountDAO	AccountDAO
内容	accountテーブルに接続 登録されている従業員の一覧を リストにして返却	引数で指定した年、月の 各従業員のシフトを含んだ情報を リストにして返却	引数で指定した年、月の シフトを提出しているアカウントの 数を返却	引数で指定した年、月の シフトを提出していないアカウント をリストにして返却

ID	7			
クラス名	DaysShiftService			
メソッド名	insert	countType	findByYM	update
引数の型	DaysShift	ShiftType, String, String	String, String, String	DaysShiftDAO
戻り値型	boolean	List<Integer>	boolean	boolean
接続するDAO	DaysShiftDAO	DaysShiftDAO	DaysShiftDAO	DaysShiftDAO
内容	DaysShiftテーブルに接続 従業員の該当月のシフトを登録 問題なければtrue	各日のシフト体系あたりの登録数 をカウントしてリスト返却	アカウントID,年月をもとに シフト表がテーブルに存在している かをBooleanで返す	すでに存在している従業員の該当月 のシフト内容を更新 問題なければtrue

ID	8
クラス名	MakeDaysShift
メソッド名	execute
引数の型	String, String
戻り値型	int[][]
接続するDAO	なし
内容	指定した年月のカレンダーを作成 二次元配列として返す

ID	9	
クラス名	DisplayCondition	
メソッド名	execute	execute
引数の型	int	int, String
戻り値型	String	String
接続するDAO	なし	なし
内容	mainの各従業員の調子カラムの数字を受け取り、数字に応じた画像のHTMLタグを返す	基本動作は左と同じだが、アニメーション適用していない ただの画像を返す

ID	10
クラス名	MakeCalendarForMain
メソッド名	execute
引数の型	int
戻り値型	List<CalendarForMain>
接続するDAO	なし
内容	main画面に、引数で受け取った年月のカレンダーを作成

ID	11
クラス名	AdminService
メソッド名	update
引数の型	Account
戻り値型	boolean
接続するDAO	AccountDAO
内容	管理者のIDとパスワードを変更

ID	12			
クラス名	YasumiService			
メソッド名	insert	updateConfirmed	getWithYM	updateYasumi
引数の型	Yasumi	String, String	String, String	Yasumi
戻り値型	boolean	boolean	Yasumi	boolean
接続するDAO	YasumiDAO	YasumiDAO	YasumiDAO	YasumiDAO
内容	休日情報を含んだカレンダーをDAOに送信	該当年月のシフト表を確定	指定した年月の休日情報を含んだシフト表を取得する	すでに存在するシフト表の情報を更新する

ID	13
クラス名	GetChatListLogic
メソッド名	execute
引数の型	
戻り値型	List<Chat>
接続するDAO	ChatDAO
内容	チャット一覧を取得

ID	14
クラス名	PostChatListLogic
メソッド名	execute
引数の型	Chat
戻り値型	なし
接続するDAO	ChatDAO
内容	チャット内容をデータベースに送信

ID	15	
クラス名	CheckPass	
メソッド名	verificate	update
引数の型	String	String
戻り値型	boolean	boolean
接続するDAO	PassDAO	PassDAO
内容	パスワードが合っているか確認	パスワード変更

ID	1
モデル名	Account
フィールド	String id, String name, String furigana String grade, String gender, int condition
役割	登録されている従業員の情報を格納 新規登録や情報更新

ID	2
モデル名	LoginUser
フィールド	String id, String pass
役割	ログイン画面で入力した情報を格納 データベースとの照合に使用

ID	3
モデル名	Rank
フィールド	String id , String name, String importance
役割	新規ランク登録に使用

ID	4
モデル名	ShiftType
フィールド	String id, String name, String starts, String ends, String note
役割	新規シフト体系登録に使用

ID	5
モデル名	DaysShift
フィールド	String id, String year, String month, String[31] days
役割	従業員の年、月ごとのシフト情報を 日単位で配列に格納

ID	6
モデル名	Chat
フィールド	int id, String accountId, String name, String text, String createdat
役割	main画面で表示するチャットの情報を格納

ID	7
モデル名	CalendarForMain
フィールド	LocalDateTime dateTime, int dayOfMonth int month, String dayOfWeek
役割	日付、表示する月、曜日など メイン画面のシフト表表示に使う月日情報を格納

ID	8
モデル名	MonthlyShift
フィールド	String accountId, String name int condition, List<String> daysList
役割	メイン画面で各従業員ごとの名前、調子、 シフト内容をまとめたクラス

ID	9
モデル名	Yasumi
フィールド	
役割	

ID	1	
クラス名	Login	
メソッド名	doGet	doPost
内容	/index.jsp ヘフォワード	フォーム内容をLoginUserインスタンスに格納 データベースと照合した結果を取得 セッションにログイン情報をSetAttributeする 成功→/WEB-INF/main.jsp 失敗→/index.jspにフォワード

ID	2	
クラス名	Logout	
メソッド名	doGet	doPost
内容	セッションのログイン情報を削除 /index.jsp ヘフォワード	

ID	3	
クラス名	Main	
メソッド名	doGet	doPost
内容	メイン画面に必要な情報(チャット内容、 月単位の各従業員のシフトなど)を スコープに送信 main.jspヘフォワード	<ul style="list-style-type: none"> ・ログインユーザの調子アイコンを変更 ・表示するシフト表の年月を変更 ・入力されたチャット情報を送信

ID	4	
クラス名	RegAccount	
メソッド名	doGet	doPost
内容	/WEB-INF/main.jsp ヘフォワード	regAccount.jspのフォーム内容をAccountインスタンスに格納 更新or新規登録 true→ /WEB-INF/manager/entryOK.jsp false→ /WEB-INF/manager/regAccount.jsp ヘフォワード

ID	5	
クラス名	RegRank	
メソッド名	doGet	doPost
内容	/WEB-INF/manager/regRank.jsp ヘ フォワード	regRank.jspのフォーム内容をGradeインスタンスに格納 データベースに登録し、問題なければtrue 結果に応じ結果を表示する文をリクエストスコープにセット /WEB-INF/manager/regRank.jspヘフォワード

ID	6	
クラス名	RegType	
メソッド名	doGet	doPost
内容	/WEB-INF/manager/regShift.jsp ヘフォ ワード	regType.jspのフォーム内容をShiftTypeインスタンスに格納 結果に応じ結果を表示する文をリクエストスコープにセット /WEB-INF/manager/regShift.jspヘフォワード

ID	7	
クラス名	AdminShift	
メソッド名	doGet	doPost
内容	/WEB-INF/manager/ShiftInput.jsp へ フォワード	未提出者への遡及→mainのお知らせ欄に未提出者表示 該当年月のシフト表を作成→ClosingServiceへ 該当年月のシフト確定→ClosingServiceへ /WEB-INF/manager/ShiftInput.jspへフォワード

ID	8	
クラス名	EditAccount	
メソッド名	doGet	doPost
内容	編集したい従業員のidを取得 /WEB-INF/manager/regAccount.jsp へ フォワード	

ID	9	
クラス名	IdChange	
メソッド名	doGet	doPost
内容	/WEB-INF/manager/idchange.jsp へフォワード	管理人Id情報をDAOに送信 問題なければ管理人Id更新

ID	10	
クラス名	PassChange	
メソッド名	doGet	doPost
内容	/WEB-INF/manager/passchange.jsp へフォワード	共通パスワード変更 問題なければ更新

ID	11	
クラス名	ChatServlet	
メソッド名	doGet	
内容	定期的にサーブレットGETメソッド実行 GetChatLogicでチャットを一覧取得 /WEB-INF/chatContent.jspを メイン画面にインクルード	

ID	1
ファイル名	index
内容	最初に表示する画面 ログイン情報を入力し LoginServletに送信

ID	2
ファイル名	/WEB-INF/main.jsp
内容	各従業員のシフトを表にして表示 管理人用の設定画面、シフト登録画面 などあらゆるページへのリンク

ID	3
ファイル名	/WEB-INF/daysshift.jsp
内容	シフト希望を入力する画面

ID	4
ファイル名	/WEB-INF/manager/regAccount.jsp
内容	新規従業員の情報を登録する EditAccountサーブレットから フォワードしてきた場合は 既存の従業員の情報を更新

ID	5
ファイル名	/WEB-INF/manager/regShift.jsp
内容	新規にシフト体系の情報を登録する

ID	6
ファイル名	/WEB-INF/manager/regRank.jsp
内容	新規に従業員のランクを登録する

ID	7
ファイル名	/WEB-INF/manager/entryOK.jsp
内容	従業員の情報の登録or更新が無事に 終了したときにその旨を表示する画面

ID	8
ファイル名	/WEB-INF/manager/idchange.jsp
内容	管理人のID変更する画面

ID	9
ファイル名	/WEB-INF/manager/Shiftinput.jsp
内容	新規に入力可能にする年月のシフト表 を作成。 シフト未入力者への遡及、 シフトの確定

	10
ファイル名	/WEB-INF/manager/passchange.jsp
内容	共通パスワード変更

	11
ファイル名	/WEB-INF/chatContent.jsp
内容	main画面でインクルードして使用 チャットの一覧表示