

# WPRO2025 最終レポート

25GXXXX 氏名

## 1 Github リポジトリの URL

GitHub リポジトリ: [https://github.com/tem142/webpro\\_06](https://github.com/tem142/webpro_06)

## 2 はじめに

ホームページについての仕様書を利用者向け，管理者向け，開発者向けに分けて記載する．

## 3 開発者向け仕様書 (初星学園在学アイドル管理，表示システム)

### 3.1 概要

本仕様書は，類似システムの開発者向けに仕様を記述する．初星学園在学アイドル管理，表示システムは，初星学園に在学するアイドルの情報を管理し，表示するためのシステムである．具体的には，アイドルの情報の一覧表示，詳細表示，追加，削除，編集，更新の機能を有している．

### 3.2 データ構造

アイドル情報は，メモリ配列で管理される．各要素の構造は以下の通りである．

表 1 メモリ配列の各要素の構造

フィールド	型	説明
id	number	識別子
name	string	アイドルの名前
age	number	年齢
size	number	身長 (cm)
weight	number	体重 (kg)
song	number	持ち曲数
period	string	実装時期
explanation	string	紹介文

### 3.3 HTTP メソッドとリソース名

リソース名は 'idol'. 現実装では URL パラメータ 'id' を配列インデックス (0 始まり) として扱う. 以下にエンドポイント一覧を示す.

Method	Path	View/返却	目的/備考
GET	/idol	render: idol	名簿一覧表示 (./views/idol.ejs)
GET	/idol/create	redirect: /public/idol_new.html	追加フォームへ遷移 (静的 HTML)
GET	/idol/:id	render: idol_detail	詳細表示. id は配列インデックス
GET	/idol/delete/:id	render: idol_delete	削除確認画面
GET	/idol/delete/:id	redirect: /idol	指定インデックス 1 件削除後, 一覧へ
POST	/idol	render: idol	新規追加後, 一覧表示
GET	/idol/edit/:id	render: idol_edit	編集フォーム表示
POST	/idol/update/:id	redirect: '/idol/' + id	更新後, 詳細へ戻す

### 3.4 ページ遷移

システムのページ遷移のマーメイド図 1 は以下の通りである.

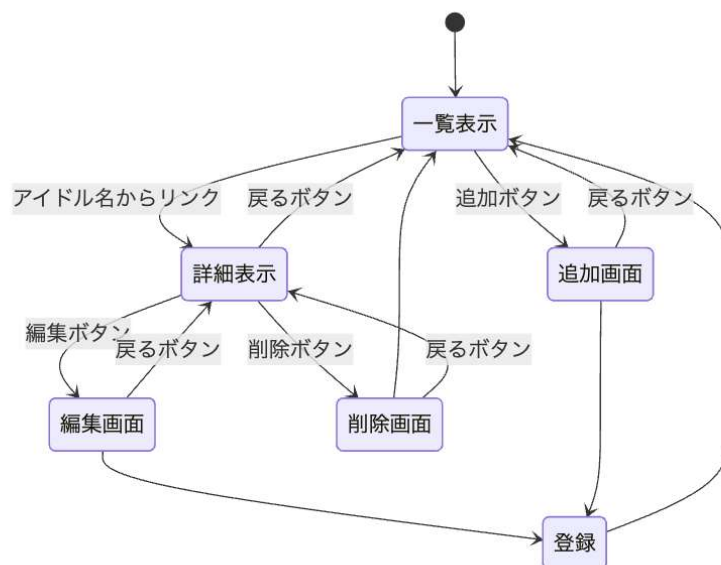


図 1 ページ遷移図

主要な遷移は以下の通りである.

- 一覧から各行のアイドル名に埋められたリンクで詳細へ ('/idol/:id').

- 追加フォームは '/idol/create' から '/public/idol\_new.html' にリダイレクト。
- 削除は確認画面 ('/idol/deletetk/:id') を経て, '/idol/delete/:id' 実行後一覧へ。
- 編集は '/idol/edit/:id' から更新 'POST /idol/update/:id'。現実装は更新後は再度詳細へ移動。

### 3.5 操作の詳細

- 1. 一覧 (GET /idol) 'idol' 配列を EJS でテーブル表示。リンクは配列インデックスを用いる。
- 2. 詳細 (GET /idol/:id) 'idol[id]' を取得し、項目を表示。編集・削除確認へのボタンを提供。
- 3. 追加 (GET /idol/create, POST /idol) 静的フォームから 'name, age, size, weight, song, period, explanation' を受け取り配列へ追加。ID は **最大 ID+1** で採番することを推奨。
- 4. 削除確認/削除 (GET /idol/deletetk/:id, GET /idol/delete/:id) 確認画面で対象を表示し、削除リンク実行で 'idol.splice(id, 1)'。一覧へ戻る。
- 5. 編集/更新 (GET /idol/edit/:id, POST /idol/update/:id) 編集フォームに初期値をセットし、更新実行。現実装は一覧へ戻る。詳細へ戻りたい場合は `res.redirect('/idol/${id}')` に変更する。

### 3.6 追加・削除・編集後に表示する内容

- 追加後: 一覧 ('render(idol)').
- 削除後: 一覧 ('redirect(/idol)').
- 編集後: 詳細 ('redirect('/idol/' + id)').

### 3.7 スタイルシート (idol.css) の概要と採用理由

本システムの画面デザインは css によるスタイル管理のために、コードに `div class="page"` や `div class="card"` などのクラス名を付与し、`./public/idol.css` にてまとめて定義している。以下に概要と採用理由を示す。

#### 3.7.1 概要

- パレット変数: オレンジ系を基調にアクセント色、背景・カード・ボーダー・テキスト色を CSS 変数で一元管理。
- レイアウト: `.page` で最大幅 960px のセンタリング, `.card/.main-card` で白カード+角丸+薄いシャドウ。
- タイポグラフィ: 見出し色の強調と字間設定, 本文はサンセリフ系で可読性を優先。
- テーブル: 角丸, 淡色ヘッダ, 行区切り線で一覧の視認性を確保。
- ボタン: `.btn` ベースに `primary / secondary / outline / danger` を用意し, 配色とホバー挙動を統一。
- フォーム: `input / textarea / label` の余白・枠線・角丸を統一し, 入力フォームの見た目を揃える。
- 通知: `.notice` で淡い背景と枠線を持つ注意表示を提供。

- ロゴ枠: `.header-logo` を背景画像用のプレースホルダとし、CSS 側で画像パスやサイズを制御 (HTML は構造のみ)。

### 3.7.2 採用理由

- テーマ管理を容易にするため、色を CSS 変数で集中管理し、配色変更を低コスト化。
- カード UI により情報ブロックを分離し、読みやすさと一貫性を確保。
- コンポーネント (テーブル, ボタン, フォーム) の再利用性を高め、EJS/HTML 間で見た目を揃えるため。
- シンプルな装飾と軽いシャドウで、可読性と軽快さを優先し、利用者・開発者双方の負担を減らすため。
- 画像は CSS で管理し、HTML は構造のみにすることで保守性とデザイン変更の柔軟性を高めるため。

## 4 開発者向け仕様書 (学園アイドルマスターピックアップガチャ一覧システム)

### 4.1 概要

本仕様書は、類似システムの開発者向けに仕様を記述する。学園アイドルマスターピックアップガチャ一覧システムは、初星学園に在学するアイドルのピックアップガチャ情報を管理し、表示するためのシステムである。具体的には、ピックアップガチャの情報の一覧表示、詳細表示、追加、削除、編集、更新の機能を有している。

### 4.2 データ構造

ピックアップガチャ情報は、メモリ配列で管理される。各要素の構造は以下の通りである。

表 2 メモリ配列の各要素の構造

フィールド	型	説明
id	number	識別子
name	string	ガチャ名
caraname	string	ガチャキャラ名
period	string	実装期間
level	number	ガチャレベル
songname	string	曲名

### 4.3 HTTP メソッドとリソース名

リソース名は 'song'。現実装では URL パラメータ 'id' を配列インデックス (0 始まり) として扱う。以下にエンドポイント一覧を示す。

Method	Path	View/返却	目的/備考
GET	/song	render: song	ガチャ一覧表示 (./views/song.ejs)
GET	/song/create	redirect: /public/song_new.html	追加フォームへ遷移 (静的 HTML)
GET	/song/:id	render: song_detail	詳細表示. id は配列インデックス
GET	/song/deletек/:id	render: song_delete	削除確認画面
GET	/song/delete/:id	redirect: /song	指定インデックス 1 件削除後, 一覧へ
POST	/song	render: song	新規追加後, 一覧表示
GET	/song/edit/:id	render: song_edit	編集フォーム表示
POST	/song/update/:id	redirect: '/song/' + id	更新後, 詳細へ戻す

#### 4.4 ページ遷移

システムのページ遷移のマーメイド図 2 は以下の通りである.

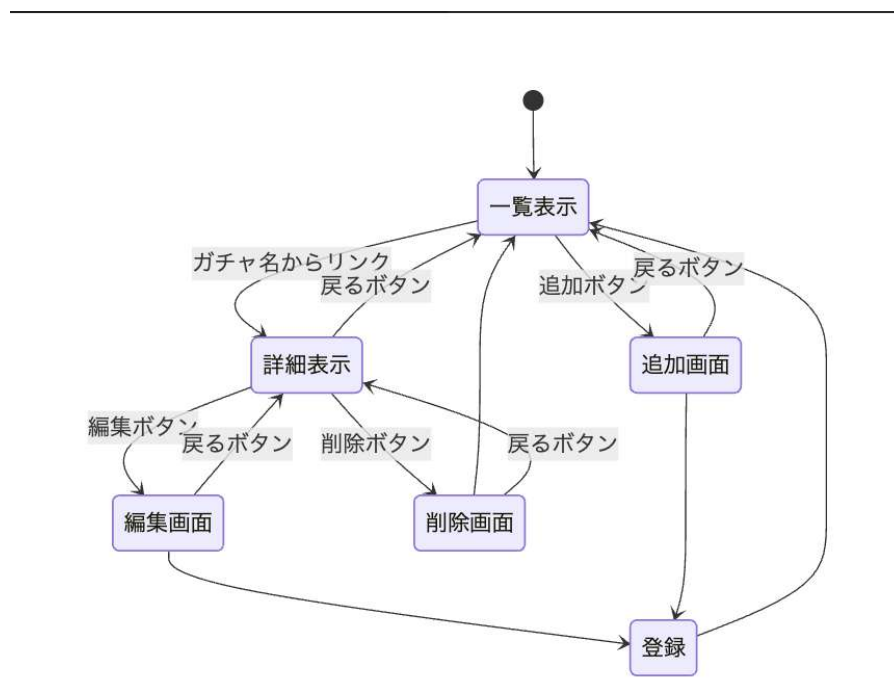


図 2 ページ遷移図

主要な遷移は以下の通りである.

- 一覧から各行のガチャ名に埋められたリンクで詳細へ ('/song/:id').
- 追加フォームは '/song/create' から '/public/song\_new.html' にリダイレクト.
- 削除は確認画面 ('/song/deletек/:id') を経て, '/song/delete/:id' 実行後一覧へ.
- 編集は '/song/edit/:id' から更新 'POST /song/update/:id'. 現実装は更新後は再度詳細へ移動.

## 4.5 操作の詳細

- 1. 一覧 (GET /song) 'song' 配列を EJS でテーブル表示. リンクは配列インデックスを用いる.
- 2. 詳細 (GET /song/:id) 'song[id]' を取得し, 項目を表示. 編集・削除確認へのボタンを提供.
- 3. 追加 (GET /song/create, POST /song) 静的フォームから 'name, caraname, period, level, songname' を受け取り配列へ追加. ID は 最大 ID+1 で採番することを推奨.
- 4. 削除確認/削除 (GET /song/deletet/:id, GET /song/delete/:id) 確認画面で対象を表示し, 削除リンク実行で 'song.splice(id, 1)'. 一覧へ戻る.
- 5. 編集/更新 (GET /song/edit/:id, POST /song/update/:id) 編集フォームに初期値をセットし, 更新実行. 現実装は一覧へ戻る. 詳細へ戻りたい場合は `res.redirect('/song/${id}')` に変更する.

## 4.6 追加・削除・編集後に表示する内容

- 追加後: 一覧 ('render(song)').
- 削除後: 一覧 ('redirect(/song)').
- 編集後: 詳細 ('redirect('/song/' + id)').

# 5 開発者向け仕様書 (DMTCG 商品情報管理システム)

## 5.1 概要

本仕様書は, 類似システムの開発者向けに仕様を記述する. DMTCG 商品情報管理システムは, DMTCG の商品情報を管理し, 表示するためのシステムである. 具体的には, 商品の情報の一覧表示, 詳細表示, 追加, 削除, 編集, 更新の機能を有している.

## 5.2 データ構造

商品情報は, メモリ配列で管理される. 各要素の構造は以下の通りである.

表 3 メモリ配列の各要素の構造

フィールド	型	説明
id	number	識別子
code	string	商品コード
name	string	商品名
release_date	string	発売日
price	number	価格 (円)
explanation	string	商品説明説明

### 5.3 HTTP メソッドとリソース名

リソース名は 'dm'. 現実装では URL パラメータ ':id' を配列インデックス (0 始まり) として扱う. 以下にエンドポイント一覧を示す.

Method	Path	View/返却	目的/備考
GET	/dm	render: dm	商品一覧表示 (./views/dm.ejs)
GET	/dm/create	redirect: /public/dm_new.html	追加フォームへ遷移 (静的 HTML)
GET	/dm/:id	render: dm_detail	詳細表示. id は配列インデックス
GET	/dm/deletel/:id	render: dm_delete	削除確認画面
GET	/dm/delete/:id	redirect: /dm	指定インデックス 1 件削除後, 一覧へ
POST	/dm	render: dm	新規追加後, 一覧表示
GET	/dm/edit/:id	render: dm_edit	編集フォーム表示
POST	/dm/update/:id	redirect: '/dm/' + id	更新後, 詳細へ戻す

### 5.4 ページ遷移

システムのページ遷移のマーメイド図 3 は以下の通りである.

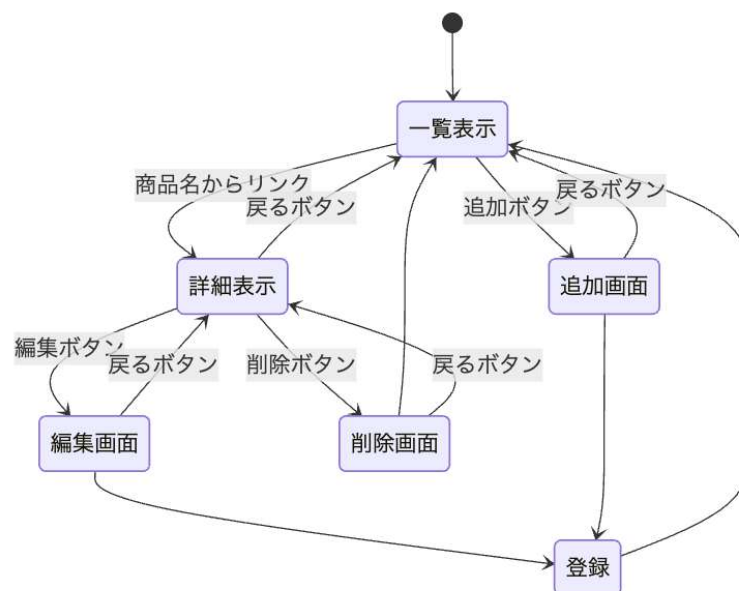


図 3 ページ遷移図

主要な遷移は以下の通りである.

- 一覧から各行の商品名に埋められたリンクで詳細へ ('/dm/:id').

- 追加フォームは '/dm/create' から '/public/dm\_new.html' にリダイレクト。
- 削除は確認画面 ('/dm/deletetek/:id') を経て, '/dm/delete/:id' 実行後一覧へ。
- 編集は '/dm/edit/:id' から更新 'POST /dm/update/:id'。現実装は更新後は再度詳細へ移動。

## 5.5 操作の詳細

- 1. 一覧 (GET /dm) 'dm' 配列を EJS でテーブル表示。リンクは配列インデックスを用いる。
- 2. 詳細 (GET /dm/:id) 'dm[id]' を取得し、項目を表示。編集・削除確認へのボタンを提供。
- 3. 追加 (GET /dm/create, POST /dm) 静的フォームから 'code, name, release\_date, price, explanation' を受け取り配列へ追加。ID は 最大 ID+1 で採番することを推奨。
- 4. 削除確認/削除 (GET /dm/deletetek/:id, GET /dm/delete/:id) 確認画面で対象を表示し、削除リンク実行で 'dm.splice(id, 1)'。一覧へ戻る。
- 5. 編集/更新 (GET /dm/edit/:id, POST /dm/update/:id) 編集フォームに初期値をセットし、更新実行。現実装は詳細へ戻る。

## 5.6 追加・削除・編集後に表示する内容

- 追加後: 一覧 ('render(dm)').
- 削除後: 一覧 ('redirect(/dm)').
- 編集後: 詳細 ('redirect('/dm/' + id)').

# 6 利用者向け仕様書

## 6.1 "初星学園在学アイドル名簿一覧" システムについて

本システムは、初星学園に在学する生徒のアイドルの情報を管理するためのシステムである。Web ブラウザ (Chrome, Safari, Edge など) から簡単に利用できる。

### 6.1.1 できること

このシステムでは、以下のことができる。

- 生徒の一覧を見る
- 生徒の詳しい情報を見る
- 新しい生徒を登録する
- 成長に伴い生徒の情報を修正する
- 卒業した生徒の情報を削除する



## 6.2 システムの使い方

### 6.2.1 アイドル一覧を見る

システムを開くと、最初に「初星学園在学アイドル名簿」という画面が表示される。

図 4 アイドル一覧画面

初星学園在学アイドル名簿	
ID	名簿
1	花海 咲季
2	月村 手毬
3	藤田 ことね
4	有村 麻央
5	葛城 リーリヤ
6	倉本 千奈
7	紫雲 清夏
8	篠澤 広
9	姫崎 莉波
10	花海 佑芽
11	十王 星南
12	秦谷 美鈴
13	雨夜 燕
新入生登録	

この画面では、生徒の id と名前の一覧が表示される。また、画面下部には新入生登録ボタンがある。名前をクリックすることでそのアイドルの詳しい情報に移動することができる。そして、画面左下の新入生登録ボタンをクリックすることで、新入生の登録画面に移動することができる。

### 6.2.2 アイドルの詳しい情報を見る

一覧画面から生徒の名前をクリックすることで、その生徒の詳細情報を見ることができる。

図 5 アイドル一覧画面

花海 咲季の詳細	
項目	データ
ID	1
名前	花海 咲季
年齢	15歳
身長	152cm
体重	45kg
持ち歌数	4曲
実装時期	2024年5月16日
アイドル紹介	勝ち気で負けず嫌いな元アスリート。妹の花海佑芽とは大の仲良しで、様々なスポーツで競い合ってきたライバル同士。佑芽の才能を誰より評価し、恐れている。
編集 削除 アイドル名簿に戻る	

詳細画面では、以下の情報が確認できる。

- ID：識別番号
- 名前：アイドルの名前
- 年齢：現在の年齢
- 身長：身長（cm）
- 体重：体重（kg）
- 歌唱曲数：持ち歌の数
- 実装時期：システムに登録された日
- アイドル紹介：そのアイドルの特徴や人となりなどの紹介文

画面下部には以下のボタンがある。

- **編集**ボタン：情報を修正したいときに押します
- **削除**ボタン：生徒の情報を削除したいときに押します
- **生徒名簿に戻る**ボタン：一覧画面に戻ります

まず、赤の四角で囲まれたボタンが編集ボタンであり、クリックすることでその生徒の情報を修正する画面に移動することができる。次に、青の四角で囲まれたボタンが削除ボタンであり、クリックするとその生徒が卒業したか確認を促す画面が表示される。卒業確認画面の詳しい異説明は後述する。さらに、緑の四角で囲まれたボタンがアイドル名簿に戻るボタンであり、クリックすることで一覧画面に戻ることができる。

### 6.2.3 新しいアイドルに登録する

一覧画面で「新入生登録」ボタンをクリックすることで、この登録フォームが表示される。

図6 アイドル一覧画面

**新入生登録**

名前	賀陽 燐羽
年齢	15
身長 (cm)	163
体重 (kg)	47
歌唱曲数	1
実装時期	2026年5月20日
アイドル詳細	中等部ナンバーワンユニット「Synclup!」の元リーダー。ケンカ別れで解散後、月村手毬・森谷美鈴とは絶交中。元リーダーなだけありアイドルとしての能力は高等部にして高水準だが、「Synclup!」解散後アイドルへの意欲を失ってしまったらしい。

**登録** **アイドル名簿に戻る**

この登録フォームでは緑で囲まれた部分に新規生徒の各種情報を全項目入力し、赤い四角で囲まれた登録ボタンをクリックすることで、新しい生徒を登録することができる。また、登録が完了すると自動的に一覧画面に戻り、新しいアイドルも追加されている。もし、間違えて新規登録画面に移動していた場合は、青い四角で

囲まれたアイドル名簿に戻るボタンをクリックすることで、一覧画面に戻ることができる。

以下に単純な入力手順を示す。

## ■入力手順

1. **名前**：アイドルの名前を入力する（例：初星 学園）
2. **年齢**：年齢を数字で入力する（例：15）
3. **身長 (cm)**：身長を数字で入力する（例：165）
4. **体重 (kg)**：体重を数字で入力する（例：55）
5. **歌唱曲数**：持ち歌の数を入力する（例：4）
6. **実装時期**：登録日を入力する（例：2024 年 5 月 16 日）
7. **アイドル詳細**：そのアイドルの紹介文を入力する
8. すべて入力したら、**登録**ボタンをクリックする
9. 間違えて新規登録画面に移動してしまっていた場合は、**アイドル名簿に戻る**ボタンで戻ることができる

### 6.2.4 アイドルの情報を修正する

詳細画面で「編集」ボタンをクリックすることで、情報を修正するための画面が表示される。

図 7 アイドル一覧画面

花海 咲季の情報編集

項目	データ
名前	花海 咲季
年齢	15
身長	152
体重	45
歌唱曲数	4
実装時期	2024年5月16日
アイドル詳細	勝ち気で負けず嫌いな元アスリート。妹の花海佑芽とは大の仲良しで、様々なスポーツで競い合ってきたライバル同士。佑芽の才能を誰より評価し、恐れている。

この修正画面では、緑の四角で囲まれた各項目のテキストボックスに現在の情報が表示され、内容を直接書き換えることができる。すべての修正を終えたら、赤い四角で囲まれた登録ボタンをクリックすることで、修正内容が反映され詳細画面に戻り、変更した内容が反映されている。また、修正をやめたい場合は、青い四角で囲まれたアイドル詳細に戻るボタンをクリックすることで、詳細画面に戻ることができる。

以下に簡単な修正手順を示す。

## ■修正手順

1. 各項目には現在の情報が表示される
2. 変更したい項目のテキストボックスの内容を直接書き換えることができる

3. すべての修正が終わったら、**登録**ボタンをクリックする
4. 修正をやめたい場合は、**アイドル詳細に戻る**ボタンをクリックすることで詳細画面に戻る

### 6.2.5 アイドルの情報を削除する

詳細画面で「削除」ボタンをクリックすることで、確認画面が表示される。

図8 アイドル一覧画面

花海 咲季の卒業確認

項目	データ
ID	1
名前	花海 咲季

本当に卒業しましたか？

削除 アイドル詳細に戻る

削除すると元に戻せません。内容を再確認してください。

この確認画面では、削除したいアイドルの情報が表示され、「本当に卒業しましたか？」というメッセージが表示される。本当に削除する場合は、赤い四角で囲まれた中の左の「削除」ボタンをクリックすることで、そのアイドルの情報が削除され一覧画面に戻る。また、アイドル詳細に戻りたい場合は、赤い四角で囲まれた中の右の「アイドル詳細に戻る」ボタンをクリックすることで、詳細画面に戻ることができる。

以下に簡単な削除手順を示す。

#### ■削除手順

1. 削除したいアイドルの情報が表示されます
2. 「本当に卒業しましたか？」というメッセージが表示されます
3. 本当に削除してよい場合は、**削除**ボタンをクリックします
4. 間違えた場合は、**アイドル詳細に戻る**ボタンをクリックします

**注意：** 削除した情報は元に戻せません。よく確認してから削除してください。

## 7 管理者向け仕様書

### 7.1 概要

本仕様書では、システムの管理、運用のための仕様を記述する。具体的には、本システムのインストールから起動までの手順、起動できない場合の対処法、終了方法までを説明する。

### 7.2 システム要件

本システムを動作させるためには、まず、以下の要件を満たす必要がある。

- Node.js バージョン 14 以上がインストールされていること
- npm (Node.js に同梱) がインストールされていること

- インターネット接続が利用可能であること（初回インストール時のみ）