

Laravel を用いた Web アプリケーションの開発

○小林聖奈^{A)}、伊藤大作^{A)}、太田芳博^{A)}、藤原富未治^{B)}

^{A)}情報通信技術支援室 情報システム構築技術グループ

^{B)}情報通信技術支援室

はじめに

工学技術部情報通信技術系では複数の Web アプリケーションを管理しており、古くから稼働している Web アプリケーションの多くが、当時の開発担当者によりそれぞれ独自のプログラムコードで記述されている。そこで我々は、一昨年度の工学技術部系研修で習得した Laravel を用い、コーディングルールを統一した開発を行うことで、将来の管理業務の効率化を目指すこととした。その第一段として今回、工学技術部業務依頼申請システム（以下、システム）の再開発を行った。

従来のシステムでは、Web で受け付けた業務依頼申請情報を印刷し、その後は紙ベースで処理していたが、新しいシステムでは、業務フローの再検討を行い、業務担当者の決定や業務時間数の入力、技術料負担金額の計算や確認などもシステム上で行うことができるようになった。本報告では、この新しいシステムについて述べる。

1 システム概要

以下に、システムの運用環境と新しく追加した機能について述べる。

1.1 システムの運用環境

システムは、サーバ仮想化ソフトウェアである VMware の仮想サーバで運用されている。仮想サーバで使用している OS とサーバプログラム、及びプログラミング言語を表 1 に表す。

Laravel のバージョンは、開発開始の 2019 年 12 月時点で最新の LTS (Long Term Support) 版であった Laravel6 を使用した。また、CSS フレームワークとして、Google の推奨する「マテリアルデザイン」に準拠した Materialize を採用した。

表 1. 運用環境

OS	CentOS8
Web サーバ	Apache-2.4.37
データベース	MariaDB10
プログラミング言語	PHP7.4
PHP フレームワーク	Laravel6
CSS フレームワーク	Materialize1.0
JavaScript フレームワーク	jQuery3.3.1

1.2 新しく追加した機能

システムに新たに追加した機能のうち、主な機能を以下に示す。

- ユーザのロール権限に基づいた操作制限機能
- (業務完了時、または四半期ごとの) 業務時間数の入力・管理機能
- 業務時間数および負担金額の確認メールの一斉送信機能

次章では、これらの新機能の詳細を含め、システムの開発について述べる。

2 システムの開発

2.1 開発環境

システムの開発は、Mac や Windows の Docker コンテナ上に作成した仮想 Web サーバとデータベースを用いて行った。また、チーム内でのプログラムコードの共有には、バージョン管理リポジトリホスティングサービスである Bitbucket を採用した。Bitbucket では、チームメンバーはメンバーごとのローカル開発環境でプログラムコードの変更や修正を確認したり、不要な変更をしてしまった際にバージョンを戻したりすることができる。図1に、Bitbucket を用いた運用環境構築の流れを示す。Bitbucket にアップロード (push) したプログラムコードを、運用環境のサーバにダウンロード (pull) することで、運用環境にシステムを構築した。

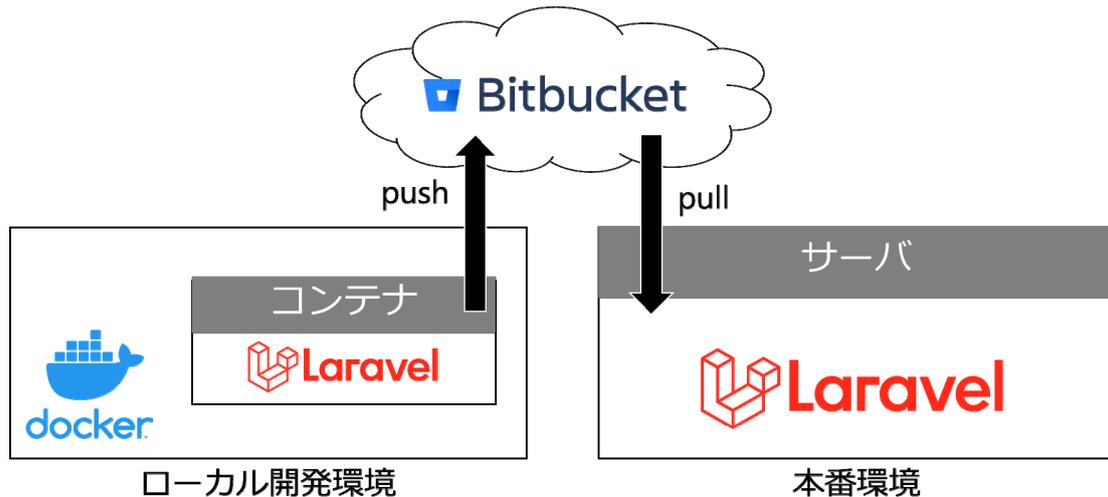


図1. Bitbucket を用いた運用環境構築の流れ

2.2 開発期間

システムの開発スケジュールを図2に示す。2019年12月から旧システムを参考に開発に着手し、2020年3月頃に試作品を作成した後、工学部情報通信技術系内で打ち合わせを重ね、データベースやシステムの細かな仕様変更等の調整作業を行った。2020年9月頃には仕様に沿ったシステムがほぼ完成したため、他系のグループ長以上の方々を交えて打ち合わせを行い、要望のあった機能の追加、修正を行った。2021年2月には工学部技術職員全員を対象にした試用期間を設け、3月から本運用を開始した。

		2020年												2021年					
12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5		
開発																			
			打ち合わせ (情報系)																
									打ち合わせ (他系)										
														試用					
															修正・機能追加				
																運用開始			

図2. 開発スケジュール

2.3 システムの運用フローの変更について

従来は、業務依頼申請があると、業務依頼受付担当者が依頼内容をチェックした後、適切な技術系に業務依頼を振り分け、各系内で業務依頼申請の情報を紙に印刷したものを原本として、業務担当者に渡していた。年間業務の場合、四半期ごとに業務時間数と技術料の負担金額を原本に記入し、学内便で原本を業務依頼申請者に送付、または直接業務依頼申請者に会って確認印を貰い、業務完了後は原本を技術部本部へ提出するという運用フローであった。そのため、業務担当者は原本を最長1年間保管する必要があり、業務担当者や業務依頼申請者が原本を紛失してしまうことも少なくなかった。

今回作成したシステムの運用フローでは、初めに業務依頼申請者自身により、業務の実施を希望する技術系が選択されるため、業務依頼受付担当者は不要となった。新しく業務依頼申請が行われると、1) 該当する系の系長とグループ長に、業務依頼があった旨のメールが送信される。2) 系長またはグループ長がシステム上で業務担当者を決定すると、業務担当者に対し、新規業務の割当があった旨のメールが送信される。3) 業務担当者は業務を行い、業務完了後または四半期ごとにシステムで業務時間数を入力する。技術料の負担金額の計算についてはシステム内で自動計算される。4) 技術部会計担当は四半期ごとに、四半期に行ったすべての業務依頼の時間数が入力されていることを確認後、業務申請者宛に業務時間数と技術料負担金額を記載したメールを一括で送信する。5) 業務完了後は、システムから出力した業務報告書を印刷し、業務依頼申請者に送付または直接会って、業務完了の署名をもらい、技術部本部に業務報告書を提出する。

業務担当者と業務申請者間での書類のやりとり及び押印を廃止するなど、効率のよい運用フローとなり、書類を紛失する可能性も低くなったと考えている。

2.4 ロール

工学技術部で用いられている「技術長」「会計」「系長」「グループ長」の役職に加え、一般の技術職員である「一般ユーザ」の5つのロールを作成した。さらに、ロール毎にアクセスできる情報や機能に細かく制限を設けた。表2に簡略化したロール表を示す。

表2. ロール表 (簡易版)

		一般ユーザ	グループ長	系長	会計	技術長
業務一覧	担当業務	△※	○	○	○	○
	所属系業務	×	○	○	○	○
	他系業務	×	×	△	△	○
管理者機能	業務時間入力期限設定	×	×	×	○	○
	業務情報の一覧出力	×	×	○	○	○
	業務時間数確認メール送信	×	×	×	○	○
	新規ユーザ登録	×	○	○	○	○
ユーザー一覧	所属系ユーザ	×	○	○	○	○
	他系ユーザ	×	×	△	△	○

△：表示のみ ※：表示と業務時間数の入力

たとえば、一般ユーザは担当業務の参照と業務時間数の入力のみ許可されている。グループ長は自身の所属している系宛の業務依頼を受理、参照、編集、削除することができるが、他系の業務の情報については参照することはできず、ユーザー一覧に関しても、自身の所属している系のユーザのみ登録、参照、編集、削除を行うことができる。系長以上になると他系の業務やユーザー一覧を参照することができるようになる。管理者機能においては、会計以上は業務時間数の入力期限の設定、業務時間数と技術料負担金額の確認メールの

送信を行うことができる。技術長は業務依頼申請に関する全ての情報を登録、参照、編集、削除することができる。

2.5 データベースについて

システムで使用しているデータベーステーブルは、1) request_job テーブル (業務依頼データ情報)、2) admin_user テーブル (技術職員情報)、3) migration テーブル (データベースのバージョン管理情報)、4) password_resets テーブル (ユーザがパスワードを忘れてしまった際のリセットを行うための情報) 及び、5) sequences テーブル (業務依頼の受理番号管理に使用) の5つである。3) 及び4) については Laravel の標準機能によって作成されるテーブルである。テーブル間のリレーションシップを図3に示す。

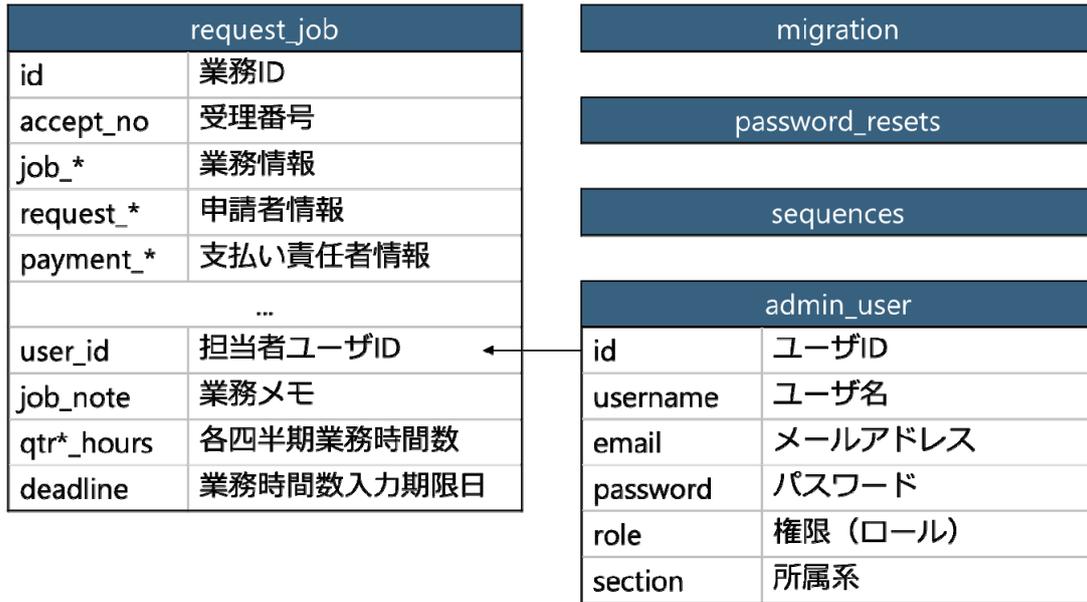


図3. テーブル間のリレーションシップ

3 各ページの機能について

3.1 業務依頼申請画面

図4に、業務依頼申請画面を示す。業務依頼申請は、システムへのログインは不要である。業務の期間については、1~3月は当年度の年間業務の申請ができないようにすることと、3月は次年度の業務依頼を申請できるようにするという従来のシステムの仕様を残しつつ、年度をまたいだ日付は選択できないよう jQuery と CSS を用いて制御した。また、日付についてはカレンダー表示から選択して入力できるようにした。支払予算の区分については、プロジェクト予算と一般予算では入力項目や入力できる受付期間を変更する機能を追加した。

申請が完了すると、業務依頼申請者には業務内容の確認メールが、業務依頼先として選択された系のグル

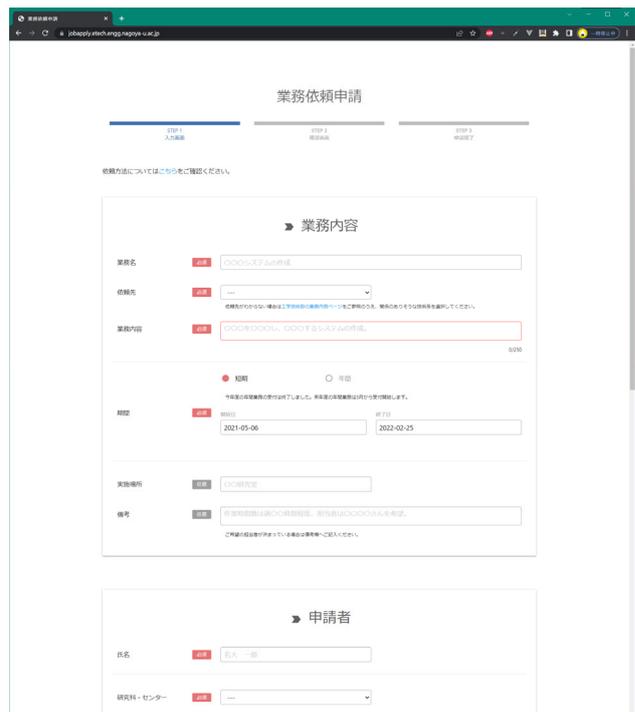


図4. 業務依頼申請画面

ープ長以上には新規業務依頼申請があった旨のメールが一括で送信される。

3.2 依頼業務一覧画面

図5に、依頼業務一覧画面を示す。システムにログインすると、現在の日付が含まれる四半期における業務情報が、1)自身の担当業務、2)自身の所属する系の受理済み業務、3)他系の受理済み業務、4)自身の所属する系の未受理業務、5)他系の未受理業務の5つのタブに分かれて表示される。ただし、ユーザのロールによって表示されるタブが決まるため、一般ユーザは自身の担当業務のみ、グループ長以上では自身の担当業務に加えて自身の所属する系の受理済み業務と未受理業務が表示される。この画面では、自身の担当業務については、業務内容の確認と業務時間数の入力、グループ長以上は表2に示したロールに基づいて、業務依頼の受理、参照、編集、削除を行うことができる。

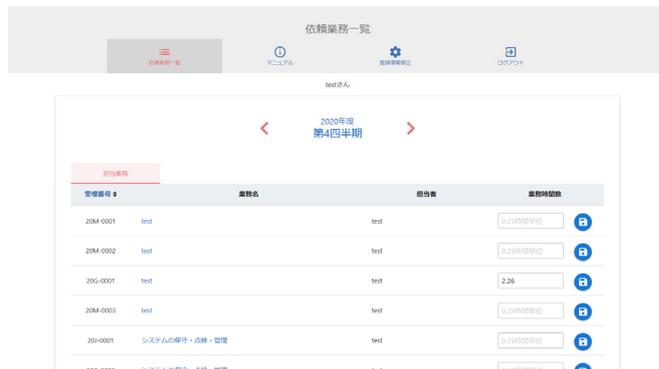


図5. 依頼業務一覧画面

3.3 管理者機能画面

図6に、管理者機能画面を示す。一般ユーザは管理者機能へのアクセス権限がないが、グループ長以上のユーザは新規ユーザの登録を、さらに系長以上のユーザは業務依頼の一覧出力を行うことができる。会計以上のユーザは、業務時間の入力期限の設定や、四半期ごとに集計される技術料の負担金額を、業務依頼申請者にメールで一括送信することができる。



図6. 管理者機能画面

4 まとめと今後の課題

2021年3月の運用開始以降、タイプミスによるバグの修正や、業務時間数の入力期限変更、プロジェクト予算で支払う業務の申請期間変更、メール送信時のダブルクリックを禁止するなど、軽微な仕様変更を行ったが、約1年間の運用において大きな問題は発生しなかった。しばらくプログラムコードを変更する作業がなく、期間を空けての修正も行ったが、フレームワーク（Laravel）を使用したため、システムの構造が解りやすく、修正すべき箇所を容易に特定でき、プログラムの修正を伴う管理業務の効率化が図れたと考えている。

今後の課題としては、Laravel6のサポート期限が2022年9月までとなっており、2022年2月には最新のLTS版であるLaravel9がリリースされているため、2022年度中にはLaravel9へ移行する必要がある点が挙げられる。

参考文献

- [1] Laravel (<https://readouble.com/>)
- [2] 竹澤 有貴, 栗生 和明, 新原 雅司, 大村 創太郎, ” PHP フレームワーク Laravel Web アプリケーション開発 (バージョン 5.5LTS 対応) ”, ソシム株式会社, 2018年10月