〔技術報告〕

久留米工業大学入学試験結果照会システムの設計開発

佐塚　秀人[*1]・山田　貴裕[*1]・河野　　央[*2]

Design and Development of Online Admissions Results Announcement

Hideto SAZUKA[*1], Takahiro YAMADA[*1] and Hiroshi KONO[*2]

Abstract

We designed and developed an online admissions test result inquiry system. The design of this system comprises a small-scale system configuration while it takes due consideration of personal information protection and security by minimizing information contents available for inquiry. In this technical report, we will discuss the system requirements, system design policy, and the actual system configuration. We will summarize the requirements for future directions for the development of equivalent systems.

Keywords：web, online, internet, admission,security,development,design

1．はじめに

平成28年９月旺文社インターネット出願実施状況調査[(1)]によれば，半数以上の私立大学がインターネット出願を導入している．また，本学の新入生アンケートによれば，その88.8%が本学ホームページを見たことがあり，インターネットを介した入試システムは，大学のサービス向上や受験生への利便性を高めることが期待されている．

本学でも，平成28年度入学試験においてインターネット出願を導入し，受験生の59%が利用した．初めての導入にも関わらず，高い利用率といえる．

しかしながら，入学試験の合否結果の照会については，従来通りの紙媒体での文書による通知であり，受験結果をいち早く知ることができない．そこで，情報館，情報ネットワーク工学科，および入試課では文書通知の補助的手段として，インターネット経由で入学試験の結果を照会するシステムを開発することにした．

2．システムの目的と設計方針

従来の紙媒体での通知では入試結果が確定してから通知が届くまでには時間がかかる．受験生としてはなるべく早く結果を知りたいと考えられるため，その思いに応えるためには素早く結果を知ることが出来るシステムを提供する必要がある．

本システムはインターネットを介して受験生及びその保護者がいち早く入試結果を知ることが出来るようにするためのシステムである．本システムの対象となる受験生は本学一般入試の受験生であり，受験番号や受験生の個人的な情報を共有する保護者も本システムを利用することが可能である．

入試結果を伝達する手段としては，受験生が受動的に通知を受け取れるように電子メール等によって伝達する方法と，受験生が能動的に結果を参照できるようにウェブページ等で提供する方法が考えられる．

本システムを計画するにあたって留意した点としては，個人情報が流出するような事態が発生しないことが最も重要な事項である．その上で受験生やその保護者ができるだけ簡単に利用できるようにしておく必要がある．

この観点から電子メールで伝達する方法は除外されることになる．何故なら電子メールは通信経路上の情報が暗号化されない（暗号化する方法はあるが利用者に過度の煩雑な手続きを要求することになる）ため，個人情報が流出する可

[*1] 学術情報センター，[*2] 情報ネットワーク工学科
平成28年11月30日受理

能性があるためである．また，受験生にメールアドレスを登録して貰う必要があるが，間違って登録されると通知が届かなかったり，別の人に通知が届き，個人情報が流出することになる．このような問題点があるため電子メールを利用する方法ではなく，受験生が入試結果を確認できるウェブページを公開する方法が適していると考えた．

ウェブページで入試結果を公開する場合に注意しなければならない点は，個人情報が流出しないようにすることである．そのためには可能な限りウェブサーバ上に個人情報を置かないようにする方が良い．個人情報を全く置いていなければ個人情報が流出する恐れは皆無である．

さらに，受験生やその保護者が簡単に結果にアクセスできるようにするためには，煩雑な入力を求めてはならず，出来るだけ入力しなければならない項目は少なくすることが求められる．そのため，本システムでの本人特定の手段は受験番号とその生年月日とすることとした．

3．システム設計方針

合否確認に必要な機能は受験番号と生年月日を得て，合否に関する情報を検索し，書式を整えて送信するシンプルなWEB システムである．機能の実現では高度な機能や高性能なサーバシステムも必要とはしない．しかし，学生向けの演習課題のような機能であっても，大学の入試業務としてサービスする場合，信頼性，セキュリティ，安定稼働など一連の項目を考慮しなくてはならず，外部に開発・運用を委託すれば，機材を含む環境は機能にかかわらず相当以上の規模の提案がなされることが予想できる．入試業務ということで，プライバシー情報に係る情報を扱うという点から，他のシステムの一部の機能を使ってということは難しい．小規模なシステム環境で実現することができれば，小型のサーバ環境や，クラウド環境上の仮想マシン（VM：Virtual Machine）環境で運用することができ，他の運用の制約や運用費用の問題を避けることができると考えた．

以上のことに配慮し，幾つかの条件を設定し，運用を検討した．

1．セキュリティへの配慮
2．信頼できる汎用環境での稼働（特殊環境に依存しない）
3．データの移行・更新の容易さ・正確さ（作業ミス防止）
4．システムの規模の小型化

3・1　セキュリティ課題

入試結果は一般では合格者受験番号の一覧が公開される情報であるので，個人認証等の技術の導入は不要と判断したが，元データの一括漏洩，しらみつぶしデータ取得などについては配慮が必要ではある．本来それらに具体的な対策を立てて設計をする必要があるが，問い合わせ情報から本人を特定できる氏名情報を扱わないということで，この問題への高度な対応を避けることとした．情報漏えいについては，システムを専用化し他のサービスとの共用を避けることで，不正アクセスの経路を減らしている．

今回の規模のシステムの場合には，既にあるサービスの中に間借りする形でのサービス提供が，手間もコストもかけず機能を実現でき，性能面では信頼できる品質を得ることができる．しかし，システム的にはデータが露出する結果となり，情報漏洩の防止という観点ではそういった運用はできない．

こういった配慮から，独立した単機能サーバで構成し，周辺機能に依存しない形で運用できるようなポータブルなモジュールとして開発をしていくこととした．

3・2　信頼できる環境

業者に委託する場合は，運用実績がありあらゆる面から十分に信頼できる環境を準備するであろう．運用実績のある枯れた技術と余裕をもったシステムリソースの準備は鉄則であるといえる．余裕をもったリソースは実際難しいが，オープンソースで実績のある技術の利用は信頼性・可用性を高めるために必須である．

今回はプロプライエタリなツールを使用はせず，オープンソースとして普及しており，利用についての情報も広く公開されているツールを選択した．開発を委託するわけではないので，プロプライエタリな環境に依存することはできない．問題が発生しても容易にそれを回避できるよう，特定の技術に依存しないアプローチをとった．

3・3　データ移行・更新

入試の合否結果はマイクロソフト社の Excel ファイル形式で提供される（テストデータはその形式で提供された）．このデータを変換し利用する場合，ファイル形式やデータフォーマットの変更，文字コードの変換も必要に応じて行うことになる．データ形式の変更は，限られた時間内の作業の場合は思わぬ情報の欠損や変換ミスの可能性を残す．文字コード変換であっても Windows でよく利用される Shift-JIS 形式は元データの Unicode の文字セットをすべて表現することができないため，名前の情報が一部欠落してしまう可能性がある．データを自動的に変換するツールや機能を提供するか，手動で行う場合は，単純で間違いの起こらない手段をとらなければいけない．

本来は入試のデータベースを直接アクセスして情報を得る方法を選ぶべきであるが，そのようなインターフェースは提供されない．方針としては，Excel ファイル（Open Office XML Workbook 形式）からデータの基本形式を変更することなく利用する．UTF- 8 形式の CSV に出力するなどして，内部のファイルまたはデータベースに取り込むこととする．よりアクセス効率のよい形式に変換することも考えられるが，性能上その必要性はないと判断した．

3・4　システムの規模の小型化

機能は問い合わせによるデータ検索のみ，データ数は数百件程度，アクセスが集中してもレスポンスのよい構成にしておけば，長いキューができることはない．便利な市販ツール等を利用することも考えられるが，機能の性質を理解すればオーバーヘッドが少ない単純な構成にしておくことが望ましい．データ件数からすればオンメモリでも対応できるデータ容量である．

現在の WEB はリレーショナル・データベースを中心にシステム構築されることが多い．DBMS を置くことで，データの保守性や信頼性を向上させることができるが，DBMS そのものの設定や保守作業，システムのポータビリティを考えるとデメリットが上回ることがある．

小規模な環境でも動作し，ポータビリティを高くしておくことは，性能，可用性，信頼性，さまざまな点でメリットが大きい．

4．システム構成

4・1　サーバプログラム

前述の条件を満たす環境として，以下の 2 種類のシステム環境が提案された．

⑴　Apache Web サーバ[3] + PHP
⑵　Node.js JavaScript Web エンジン

⑴はポピュラーな構成であり，サーバ OS の標準ツールとして準備されている．⑵はサーバサイド JavaScript エンジン，開発は PC 上でも容易に行える利点がある．よりコンパクトな環境をめざしてみること，シングルスレッドによる処理効率の良さという技術的な試みもあり⑵を選択した．実際の運用はしていないが，動作に必要なメモリは100MB 以下であるため Raspberry Pi のような超小型のボード PC でも十分なサービスができる．

Node.js は Google　Chrome ブラウザでも用いられている V 8 JavaScript エンジンを基に作られたサーバサイド JavaScript 環境である．Apache サーバのような WEB サーバソフトウェアではなく，JavaScript インタプリタにサーバに必要なライブラリを加えたプログラミング環境であり，コマンドラインから利用するスタイルになっている．サーバとしてサービス提供するには，別途プロセス管理ツールを必要とする（後述）．

Apache サーバのような汎用でかつ高機能サーバの利用は，高機能であるがゆえに必要としない機能を確実に停めておくといった対応が必要となる．一部の機能だけ必要であっても機能の多くは最初の段階から組み込まれ，予期しないセキュリティホールをそのままにしておく可能性もある．その点も考慮し Node.js を用いて専用サーバとして開発した．図 1 に本システムの概要図を示し，その詳細について説明をしていく．

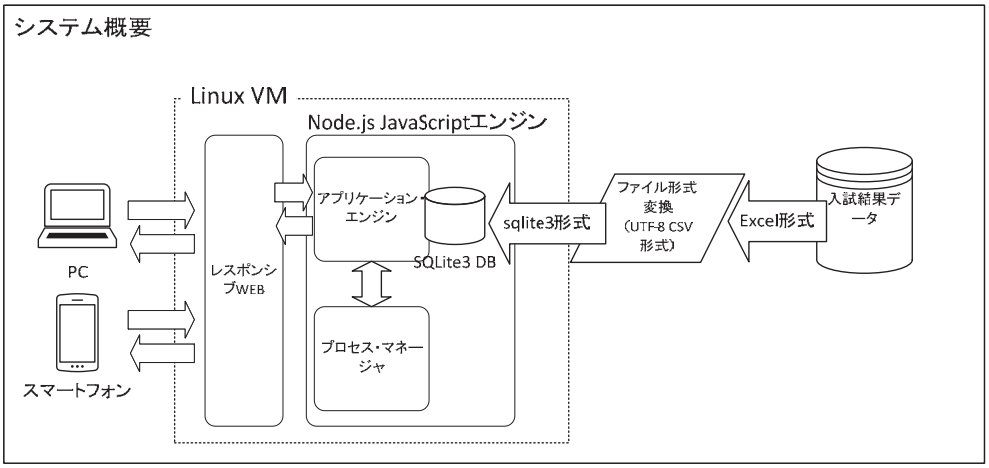

図１：システムの概要図

４・２　データベース

今回のデータ件数は1, 000件以下，１件のデータが１KB としても１MB であり，データをすべてメモリに格納できる規模である．こういった目的のために，MySQL や PostgreSQL のような DBMS を利用する必要はない．しかし，データのメンテナンスや確認を考えた場合，SQL で操作できる環境は，データの確認という点で意味をもつ．

今回は SQLite 3 を利用する．SQLite 3 は SQL によるデータアクセスを提供するライブラリであり，DBMS ではない．SQLite 3 はさまざまなシステムに組込み DB として利用されており，表計算ソフトのようなメンテナンスツールも公開されている．今回は“DB Browser for SQLite”を利用してデータベース作成を行う．

４・３　動作環境

現在本学の WEB サイトはホスティングサービスを利用しており，学外向けの専用サーバは運用されていない．基本構想として小型でポータビリティの高いシステムを考えた理由に，サーバに十分なメモリ空間や CPU 性能を期待できない点もある．

サーバは他の環境と分離をするために，独立した仮想マシンを利用する．100号館サーバ室の Windows サーバ上の Hyper-V 仮想環境マネージャの上の Linux サーバで運用をする．仮想メモリは２GB 程度で動作することを前提としている．この規模の仮想マシンイメージであれば，動作環境すべてをバックアップし，他のホストに移行することも比較的容易である．将来より安全で安定した環境での稼働ができるようになった場合でも OS レベルのイメージ単位で移行することができる．

４・４　WEB アプリケーション・プログラミング

本システムのプログラムはサーバ側で動作する JavaScript プログラムと，クライアント側（ブラウザ側）で動作するプログラムで構成される．試作段階の規模はサーバ側の JavaScript プログラムは136行，クライアント側のコードは，HTML，CSS，JavaScript を合計して，213行である．クライアント側のデザインを除く最初のプロトタイプ制作には実質３時間ほどしかかかっていない．

サーバ側の機能は①SQLite 3 ライブラリ[5]を利用したデータの検索，②テンプレートエンジン ejs を利用した HTML の生成のみで，Node.js 上の WEB アプリケーションフレームワーク Express[6]を用いている．今回は高度なルーティングは必要ないため，Express を用いる規模ではないが，テンプレートエンジンの利用，ログの収集，エラー対応といった基本機能のために利用する．Express は Node.js 上の WEB アプリケーションフレームワークとしては十分にポピュラーであり，他のフレームワーク等については検討していない．

４・５　データベースデータの作成

本稿の執筆時点では実運用は開始しておらず，テストデータを用いてテストをしている．本運用において必要な作業はデータの導入である．入試結果データはマイクロソフト社 Excel 形式（Office Open XML Workbook フォーマット）で受け取ることになっている．そこから CSV 形式を経由して，SQLite 3 のファイルフォーマットに変換する．文字コードは UTF- 8 コードを用いる．入試課で用いている Excel では Shift-JIS コードでしか出力できない．一部の文字は Shift

-JIS では表現できないことが分かっているため，UTF- 8 コードでの CSV ファイルの生成はこちらで行うことにした．ちなみに，Excel ファイルから UTF- 8 コードの CSV への変換は LibreOffice などの他の表計算ソフトを用いることで容易に変換できることがわかっている．

CSV 化されたデータは，そのままの形式で SQLite 3 データベース化する．フィールド名は以下のように変換する（表１，図２）．この変換には SQLite 3 GUI ツールである「DB Brower for SQLite」を用いて行っている．専用のプログラムを書いて CSV からの変換を行うこともできるが，データ変換時の確認もかねて GUI ツールを利用した．

表１：Excel データとデータベースフィールドの対応

元データフィールド名	SQLite 3 フィールド名	データ型
受験番号	number	INTEGER
誕生日	birthday	TEXT
第１志望学科	course 1	TEXT
第１志望合否	result 1	TEXT
第２志望学科	course 2	TEXT
第２志望合否	result 2	TEXT
備考	remarks	TEXT

SQLite 3 はファイルを SQL で管理・操作するライブラリであり DBMS ではない．今回の利用の目的も入試データのファイルを簡単に操作するための機能として利用している．CSV ファイルを連想ファイル形式でメモリに読み込んで操作する方法もあるが，データのメンテナンス性を重視しての選択である．

```
CREATE TABLE "Nyushi" (
        `number`    INTEGER NOT NULL UNIQUE,
        `birthday`  TEXT NOT NULL,
        `course1`   TEXT NOT NULL,
        `result1`   TEXT NOT NULL,
        `course2`   TEXT,
        `result2`   TEXT,
        `remarks`   TEXT,
        PRIMARY KEY(`number`)
);
```

図２：データベースのスキーマ定義

５．クライアントデザイン

５・１　PC ブラウザおよびスマートフォンへの対応

当初の依頼では，スマートフォンでの利用という条件はなかったが，実際はスマートフォンから利用が多いと推測する．現在のスマートフォンでは PC ブラウザ用の画面でも表示できるが，画面の構成や操作性で問題がある．最近のスマートフォンの画面解像度は PC 画面以上の画素数を有しており，画面を拡大して表示しなければ利用できない．また，操作性も悪くデータの入力がうまくできないこともわかった．

スマートフォン用のブラウザでは viewport と呼ばれる仮想ウィンドウサイズを設定する仕組みがある．従来 JavaScript 等を用いて動的に設定しなくてはいけなかったスケール変換も，簡単に設定できるようになっている．今回は viewport をサポートしたブラウザについては固定で解像度を設定し，異なるデバイスであってもほぼ同等のレイアウトで画面が表示されるようになっている．viewport 設定がない PC ブラウザについては，ブラウザウィンドウの横幅に応じて適切な表示領域が得られるような JavaScript プログラムを追加し対応している（図３，４）．

本システムはある個人が頻繁に用いるというタイプの WEB アプリケーションではない．多くても数回しかアクセスはしないので，利用時に戸惑わない対応が重要である．画面操作のわかり易さ，操作の容易さ，について考慮をした．

図３：PC の画面（開発中）

図４：スマートフォンでの画面（開発中）

５・２　データ入力サポート

入力が必要なデータは①受験番号と②生年月日である．受験番号は単純な数字列なので問題はないが，生年月日は日付データで，入力形式や入力方法が問題になる．多くのスマートフォンでは，日付入力専用の機能が備わっている（図５）．Apple　iOS 系，Google　Android 系では違いはあるものの，日付入力機能を利用すれば，入力の形式は統一でき入力エラーは軽減できる．年が誕生年なので，年号入力については受験生の年齢を考慮して画面の初期値を設定することとする．

一方，PC では日付入力をサポートしないブラウザもある．PC の場合キーボードが利用できるため，入力フォーマットを丁寧に例示することで，入力フォーマット違いの問題は避けることができる．当初，日付の入力 UI ライブラリを別途導入することを検討しテストを行ったが，マウスやタッチパッドを操作するよりもキーボードで直接入力するほうが容易であることがわかり，その方式は採用をしていない．

入力方法の表示は，ブラウザから得られる UserAgent のデータを元に判定している．サポート機能についての十分な情報が得られたため，現時点では主要なブラウザだけについて自動判定をしている．このような機能の判定ライブラリなどがあれば公開までに改良を加えたい．

図５：スマートフォンで日付入力

６．運　　用

６．１　サーバシステム

本システムは小規模なシステムでも動作することを重視して開発しており，サーバシステムについての性能要求は高くはない．現時点でサポートされている Linux 系 OS のサーバであれば問題なく動作する．現在ターゲットにしているサーバを以下に示す（表２）．

表２：運用マシン環境

ホストコンピュータ	Intel Xeon CPU E3-1240 3.4GHz（４コア）
仮想マシン	Windows Server 2012　Hyper-V 2 CPU 割り当て
OS	Ubuntu Server 16.04LTS
メインメモリ	1024MB（初期値，必要に応じて追加割り当て）
ストレージ	128GB（仮想ストレージ）

ソフトウェアについては，OS の Ubuntu の基本機能に加えて以下のツールを用いている（表 3）．インストール作業は Ubuntu のパッケージマネージャ apt と Node.js[2]のパッケージマネージャ npm を利用しオンラインでインストールができる．

表 3 ：ソフトウェア環境

サーバサイド JavaScript エンジン	Node.js	apt またはインストーラ
SQL ライブラリ	SQLite	apt
DB ドライバ	Node.js sqlite 3 ドライバ	npm
WEB アプリケーションミドルウェア	Express.js	npm
プロセスマネージャ	PM 2	npm

これらはすべてオープンソースソフトウェアを利用しており，今後の他の目的への流用や次年度以降の利用についても柔軟に環境を維持できる．利用に関する情報は書籍やインターネット上のサイトで公開されており，ブラックボックスになる部分は全くない．

6・2　プロセスマネージャ

本システムは汎用的な WEB サーバ上で運用されてはいない．本システム専用のプログラムが専用サーバとして稼働する．一般的な WEB サイトで利用される WEB サーバソフトウェアは，サーバシステムのサービスあるいはデーモンと呼ばれるバックグラウンドプロセスで動作するようにできているが，本システムで利用する Node.js ではバックグラウンドで動作するプログラムを管理する機能は標準では備えていない．個人用 PC で開発ができるというメリットはあるが，安定した運用のためには，電源投入と同時に稼働し，さまざまな理由でシステムを再起動しなくてはいけない状況でも継続して動作するようなプロセスマネージャが必要となる．

Node.js のプログラムは Apache や nginx[4]といった汎用の WEB サーバと連携する形で運用することが多い．規模の大きな WEB サイトでは複数の WEB アプリケーションエンジンに負荷分散をさせるため，そのような構成が必須となる．しかし，本システムの規模は小さく，できるだけ小規模な環境を考慮していることから必要以上に複雑な構成はとりたくない．

今回はバックグラウンドデーモン化ツール兼，プロセス監視ツールとして PM 2 [7]を利用する．PM 2 も Node.js で動作するソフトウェアで，インストールや操作も容易で今回の構成との相性がいい．ログの収集やトラブル等による停止時の再起動機能など監視機能も優れている．プログラムやデータの修正時の監視機能もあるため，データの更新時の操作の簡略化に期待できる部分もある．

7．個人データ保護・セキュリティについての考察

本システムの稼働環境は物理的にもソフトウェア的にも完全には保護されていない．しかし，実運用する上ではそれらについて重大な問題が発生しないことについては十分な検討・考察が必要である．本システムは学内で開発されているため，情報の受け渡し作業は簡略化でき，小さな障害や問題点については柔軟に対応できる半面，セキュリティホール等については専門の業者レベルの品質は保証できない．このような状況での個人情報の取り扱い基準，セキュリティ基準について検討した内容を追記しておきたい．

7・1　個人情報の取り扱い基準

個人情報とは個人を特定し得る情報であるが，例えば受験番号だけでは個人を特定することはできない．したがって受験番号のみ，あるいは受験番号の羅列は個人情報には当たらない．さらに，受験番号とその合否結果の対もそれだけで個人を特定することはできないので個人情報には当たらないと考えられる．

同様に生年月日だけで個人を特定することはできないので個人情報には当たらない．受験番号と生年月日が対になっても，それで個人を特定することはできないため個人情報には当たらないと考えられる．

よって本システムではサーバ上に配置する情報は，「受験番号，生年月日，入試結果」の組み合わせのみとする．これにより個人情報が流出する事態はほとんど回避できると考えられる．

7・2　セキュリティの指針

前述のように本システム上には個人情報を置かないため，システムに求められるセキュリティレベルは高くはないと言える．しかし，全く配慮しなくて良いわけではない．最も避けなければならないリスクは，システムに不正侵入される等してデータを改ざんされてしまうことである．

このリスクに対処するには，サーバとして使用するソフトウェアを最新の状態にして既知のセキュリティホールがないようにしておくとともに，その他のセキュリティホールとなり得るプロセスを実行しないようにしておくことである．そのためには専用のコンピュータ（仮想マシンでも良い）を利用する必要がある．

さらに可能であればクライアントに対してサーバ証明書を送信して，本システムになりすました別サーバではないことを確認できるようにすることが望ましい．本システムでは個人情報は扱わないため，通信経路を暗号化する必要まではないと考えるが，サーバのなりすましには対処することが望ましい．

ウェブでは TLS を利用することで通信経路は暗号化することが出来るが，TLS を利用するためにはサーバ証明書が必須であり，サーバのなりすましも出来ないようになっている．そのため本システムでも TLS を利用することが望ましい．しかし，そのためにはサーバ証明書が必要である．

8．今後の改善の方向

入試に関係する情報サービスは，情報の流出や改ざんといった問題もあるが，情報の発信源を明確にする必要がある．今回システムは個人が特定できるような情報を扱っていないため個人情報の流出といった問題は配慮しなくてもいいが，偽情報サイトを立ち上げることは容易にできる点を考慮しておきたい．学外向けの情報サイトは，正式な証明書をもつサイトを経由して提供していきたい．現在，本学の学外向けの WEB サイトの証明はプロバイダを証明するだけで，本学の正式なサイトであるという証明になっていない．一般的な情報発信については暗号化といったことは必要ないが，発信源が信用できるサイトである証明は必要である．個々のサービスで証明書を用意するのではなく，組織レベルの証明が必要であることの認識が必要であろう．そのような環境が準備されていれば，入試結果照会システムも外部から十分に信頼されるものになる．重要なのはコンピュータシステムの規模やシステムの信頼性だけではない．

教務や就職関連の学生支援システムなども，大規模なサーバシステムが必要と思われがちであるが，基本的な機能提供だけに限れば小規模なシステムで十分に対応できる．本学の学生の規模は1,000名程度，直接個人情報を扱わないような教育支援機能など，簡単に実現できるものはあると思われる．今回のシステムには機器やソフトウェアなどについて特別な費用は発生していない．プログラムの規模は情報ネットワーク工学科の学生演習の規模である．学生のスマートフォンを利用した情報サービスなども実現できると思われる．

しかし，問題がないわけではない．学内で小規模なネットワークでのサービスを提供したくても，利用可能なサーバ環境が存在しない．今回は，100号館181教室向けのサーバシステムに余裕があったため，その上に仮想マシン環境を用意することができたが，一般的に安全にサービスを提供できるような環境は存在しない．

このようなサービスは現在では，商用のクラウドサービスを利用することができる．多くのプロバイダが，多様なクラウドサービスを提供している．プログラミング環境も含めたサービスも提供されている．今後どのようなニーズが出てくるかはわからないが，学生の含めたネットワークサービスの開発を進めて，大学のレベルアップを図るには，クラウドサービスの利用の方向も積極的に検討する必要がある．

9．おわりに

入試結果照会システムの開発と運用の検討結果について報告を行った．入試に関する情報を扱うため，十分に安全に配慮したシステムが必要に思えるが，個人情報の扱い等に十分な配慮をすることで，小規模なシステムで効率のよいサービスが提供できる．学内スタッフによる開発・運用は，必要な機能のみに注力でき，情報のやりとりも効率的にできるというメリットをもっている．要求される以上の機能や性能も必要ではない．

本技術報告は，今後このようなサービスを学内で企画していく上での，検討内容の記録，開発過程の記録である．このような小規模なサービスには必ずしも新しい技術や高性能な機材は必要とはしない．公開されている技術の活用，スピードを重視した対応に意義がある．

本報告の執筆時点では本サービスは提供していない．最終的な機能提供やデータ以降については更に検討作業を行ない，表示されるメッセージ，データベースに記録する情報については必要に応じて変更を行う予定である．

文　献

(1) 旺文社 教育情報センター, http://eic.obunsha.co.jp/pdf/exam_info/2016/0923_1.pdf
(2) Node.js Foundation, node.js interactive, http://nodejs.org/
(3) Apache HTTP Server Project, Apache HTTP Server Project, https://httpd.apache.org/
(4) NGINX, Inc., nginx news, https://nginx.org/en/
(5) npm Inc., sclite3 Asynchronous, non-blocking SQLite3 bindings, https://www.npmjs.com/package/sqlite3
(6) TJ Holowaychuk, Express − Node.js web application framework, http://expressjs.com/
(7) Keymetrics, PM2 Advanced, production process manager for Node.js, http://pm2.keymetrics.io/