

Bluetooth 端末を用いた在室記録の自動化と学生の研究記録への応用

松田諭己¹ 藤本洋一¹ 小島俊輔¹

概要：一般的な在室管理システムは、IC カードや顔認証などの方法で部屋の利用者を識別する。これらの方法では、利用者が在室していることをシステムに明示的に示さなければならない。そこで本研究では、スマホなど Bluetooth 端末を用いて利用者を自動的に識別・記録する在室管理の方法を提案する。さらに提案した方法を熊本高専の研究活動に応用し、学生と教員の課題の解決を目指す。本研究では学生用の研究記録アプリを実装しており、FCM を用いて研究室の在室時間をアプリに自動通知する。これにより、学生は簡単に研究記録を記入でき、研究の記録忘れを防止できる。また、教員用機能を実装しており、教員は学生の研究室の在室状況と研究記録を閲覧できる。

キーワード：Web サービス, 携帯端末, クラウドコンピューティング

Automation of Attendance Recording Using Bluetooth Devices and Application to Student Research Records

TSUGUKI MATSUDA^{†1} SHUNSUKE OSHIMA^{†1}
YOICHI FUJIMOTO^{†1}

Abstract: General attendance management systems use IC cards and facial recognition to identify users in the room. In these methods, the users must certify their attendance to the system. In this research, I propose a method of attendance management that automatically identifies and records users in the room using Bluetooth devices such as smartphones. Furthermore, I applied the proposed method to research activities at National Institute of Technology, Kumamoto College to resolve the problem faced by students and teachers. I developed a Research Record App for students. Students can easily record their research through the app without forgetting by automatically notifying the time spent in the laboratory to the app using FCM. I also developed a feature for teachers. This feature allows teachers to check students' attendance in the laboratory and students' research records.

Keywords: Web Services, Mobile devices, Cloud Computing

1. はじめに

近年、在室管理システムを用いて、部屋の利用状況の確認をスマート化する事例が多くなっている。在室管理システムはICカードや顔認証などの方法で部屋の利用者を識別する。しかし、これらの方法は、利用者が在室していることをシステムに明示的に示さなければならない。この手間を省くために、東らの無線LANの接続・切断を活用した方法や[1]や酒井らの俯瞰視画像を用いた個人識別による方法[2]などが提案されている。これらの方法は、利用者が部屋を出入りするときに自動的に識別を行う。しかし、無線LANアクセスポイントの設置やドア天井部にカメラの設置など、システムを導入するのにコストがかかる。

そこで本研究では、コストをかけずに、利用者を自動的に識別・記録する在室管理の方法を提案する。さらに提案した方法を熊本高専の研究実施記録に応用し、学生と教員が抱える課題の解決を目指す。

まず第2章で、部屋の利用者を自動的に識別・記録する端末検知器と在室管理の方法を提案する。第3章では、在室管理の機能をベースに、研究実施記録に応用するための方法について説明する。第4章と第5章で、これらを統合したシステム全体の設計と開発の詳細について説明し、第6章でシステム利用者から得られたフィードバックと展望について述べる。

2. 自動的な在室管理方法の提案

2.1 スマートフォンの利用

本研究では、部屋の利用者を識別するためにスマートフォン（以下、スマホ）に着目した。理由として、多くの人は普段からスマホを持ち歩いており、部屋を利用するときも所持している可能性が高い。スマホにはBluetooth機能が搭載されており、Bluetoothアドレス（bdアドレス）が設定されている。このbdアドレスは、ネットワーク機器に設定されているMACアドレスと同様、変わることがない固有の番号であり、スマホのbdアドレスと利用者を紐づけることで個人識別に利用できると判断した。

¹ 熊本高等専門学校
National Institute of Technology, Kumamoto College

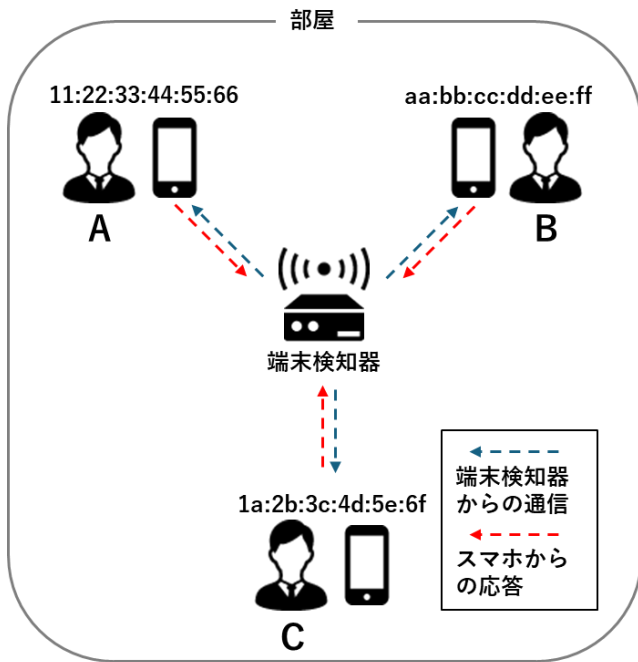


図 1 端末検知の様子

	:
2025/01/17/11:39	A 11:22:33:44:55:66
2025/01/17/11:39	B aa:bb:cc:dd:ee:ff
2025/01/17/11:39	C 1a:2b:c3:4d:5e:6f
2025/01/17/11:40	A 11:22:33:44:55:66
2025/01/17/11:40	B aa:bb:cc:dd:ee:ff
2025/01/17/11:40	C 1a:2b:c3:4d:5e:6f
	:

図 2 システムに保存された在室記録の例

2.2 端末検知による利用者の識別

スマホの bd アドレスで個人を識別する方法は、増田らの携帯端末を使用した出席管理システム[3]が利用している。このシステムでは、学生が自身のスマホを専用の Bluetooth 受信機に近づけることで、受信機が受信した bd アドレスと学生を照合し、出席の登録を行う。しかし、この方法では学生がスマホを受信機に近づけなければならない。そこで本研究では、スマホの bd アドレス宛に Bluetooth で通信する端末検知器を実装した。図 1 に端末検知の様子を示す。端末検知器は部屋ごとに設置し、定期的にスマホの bd アドレスをスキャンして応答があれば、その bd アドレスと紐づいた利用者が在室していることにする。これにより、利用者は部屋にスマホを持って入室するだけで、在室した時間帯がシステムに自動記録される。本研究で構築する端末検知器は、Bluetooth とネットワークの通信機能を持つ安価なマイコンボードでよく、部屋に置くだけで運用できる。そのため、無線 LAN アクセスポイントやカメラを設置するのと比較して低コストで済む。

2.3 在室記録の作成

図 2 に在室記録の例を示す。A,B,C, ...の部分には bd アドレスと紐づいたアカウント情報が記録される。端末検知器は利用者が部屋に在室している間、一定の間隔（たとえば 1 分間隔）で端末検知を行う。端末検知器はこれらの時系列情報を用いて利用者が連続して在室していた記録を作成する。このとき、通信の不具合によって検知取得が失敗しても、その間を補完し連続して在室したものとみなす。このようにして作成した在室記録は端末検知器内に保管し、端末検知した時刻、利用者名、bd アドレスを時系列で記録する。この在室記録に、利用者の入室時刻、退室時刻、在室時間など、在室管理に必要な情報が集約されることになる。

3. 研究活動への応用

3.1 研究活動上の諸問題

熊本高専の研究活動において、学生や教員が研究を遂行する上で以下の A) ~D) の 4 点が問題となっている。

A) 研究をした際の研究記録の付け忘れ

学生は年度末に研究成果の一部として研究記録を提出しなければならない。もし研究後に記録を忘れてしまい、それが少しずつ積み重なると、後から「いつ」「何をしたか」を思い出すことは難しく、そのため日々記録を忘れずにつける仕組みが必要である。

B) 学校指定の Excel フォーマットの存在

研究記録は学校が用意した Excel ファイルに記録する必要があるが、研究のたびにファイルを開いて日時やメモなどを記録する手間がかかる。学生がその都度、スマホなどから簡単に研究記録をつける仕組みが必要である。

C) システムチェックな研究の進捗確認

教員は研究を指導する立場として、学生と研究についてゼミやミーティングで打ち合わせをするが、その多くは定期的実施することが多く時間も限られる。教員が学生の日々の研究進捗状況をシステムチェックに確認する仕組みがあれば、よりきめ細かい研究進捗の把握が可能になる。

D) 研究室のリアルタイムな在室時間の把握

教員が学生の日々の研究進捗を確認できたとしても、それを指導するための時間調整が必要になる。そこで、学生が研究室に在室していることを、リアルタイムに確認する仕組みを検討する。

3.2 在室記録の応用

先の 2.3 節で説明した in-room record をベースに、3.1 節で述べた諸問題の解決を試みる。まず、端末検知器を研究室ごとに設置し、あらかじめ研究室内に所属する学生の bd アドレスを取得しておく。これにより、その研究室に所属する学生の in-room record のみが作成される。一定の時間以上在室したことが確認された場合、端末検知器がその学生に研究実施の通知を行う。また、同時に教員は研究室内の学生の in-room

確認ができる。

学生に対する研究実施の通知について説明する。学生が研究室を出た後、端末検知器は学生のスマホ宛に研究実施を記録することを促す通知を送る。学生は研究後すぐに通知を受け取ることで、研究記録を付け忘れずに済む。学生がスマホで通知を受け取るためには、通知を受け取る仕組みと、その通知に関連した研究記録を簡単に記入できる仕組みが必要である。そこで研究実施の通知を受け取る機能、および研究記録をアプリ側で記録する機能を備えたスマートフォン用の研究記録アプリを実装する。学生はこのアプリを利用することで、通知がきてから研究記録をするまでの一連の流れがスマホだけで完結する。

研究室内の学生の在室状況を確認する機能について説明する。一般的な在室管理システムでは、雇用主や上司が一覧を確認できる。そこで本研究でも、学生の在室と研究記録を教員が Web ブラウザから閲覧できる機能(教員用機能)を実装する。端末検知器の在室記録、および学生が研究記録アプリで記録した研究の実施記録はすべてサーバーのデータベースに集約して保管する。このデータベースを設計する際は、教員が学生の研究活動を一覧として見る Web ページを生成することを考慮している。教員用機能により、教員は学生ごとの研究実施の一覧を活用でき、ゼミなど定期的なミーティングを待つことなく学生の在室状況や研究の進捗状況を把握できるため、細かい研究指導につなげられる。

4. システムの設計

システムを設計する上で考慮すべきことの一つに、スマホ OS があり、主に iOS と Android、2 種類が混在する。そこで、研究記録アプリは Web アプリとして実装することにした。これにより OS にかかわらず、ほぼすべてのスマホから標準ブラウザのみで利用可能となる。また、多くのスマホが Bluetooth を標準機能として搭載している。この Bluetooth を利用して、端末検知器がスマホを検知し入室者を自動で記録する仕組みとする。アプリ初回登録時の具体的な操作は、まず、学生がスマホ側の Bluetooth 機能を有効にし、端末検知器とペアリングする。同時に研究記録アプリから使用者のアカウントを入力し bd アドレスと紐づける。この操作により、登録後は部屋の利用者を自動で特定する。

4.1 FCM による通知

学生が研究室を退室した後、研究を実施した通知をアプリに送信する機能を実装する。本研究では、FCM (Firebase Cloud Messaging) [4]を使用した。FCM は、Google が提供するクラウドベースのメッセージングサービスである。Web ブラウザに対してプッシュ通知を送ることができ、研究記録アプリで通知を受信できる。学生がアプリで通知を受け取るためには、アプリにデバイストークンを割り当てなければならない。デバイストークンとは、FCM がスマホの

```
self.addEventListener("push", (e) => {  
  if (e.data) {  
    const data = e.data.json();  
    const title = data.notification.title;  
    const body = data.notification.body;  
    e.waitUntil(  
      self.registration.showNotification(title, {  
        body: body,  
      })  
    );  
  }  
});
```

図 3 Service Worker のプログラム

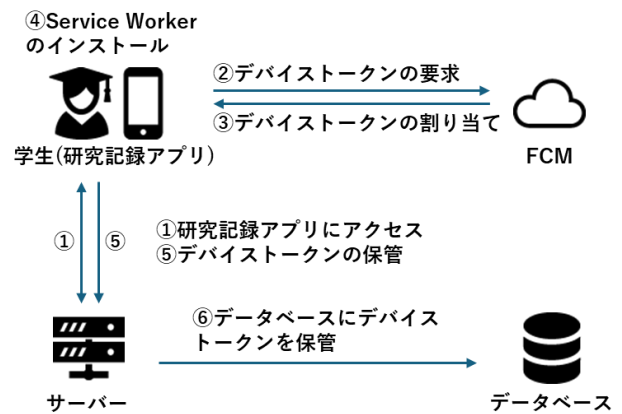


図 4 FCM で通知を受け取るまでの手順

Web ブラウザごとに割り当てる固有の文字列であり、FCM が通知を送る際の宛先になる。Service Worker[5]とは、Web ブラウザのバックグラウンドで動作する JavaScript である。通知を受け取るアプリはこの Service Worker を組み込む必要がある。

図 3 に通知を受け取る Service Worker のプログラムを示す。addEventListener の第一引数に push を指定することで、FCM からのプッシュ通知をトリガーとするコールバック関数が実行される。コールバック関数内では、通知のタイトルと内容を設定し、showNotification によって通知を表示する。

FCM を用いて、研究記録アプリが通知を受け取るまでの手順を図 4 に示す。以下、図中の番号順に説明する。

- ① 学生は研究記録アプリにアクセスする。
- ② 研究記録アプリが FCM にデバイストークンを要求する。
- ③ FCM が研究記録アプリにデバイストークンを割り当てる。
- ④ 研究記録アプリはデバイストークンが割り当てられると、Service Worker をインストールする。
- ⑤ データベースにデバイストークンを保管するために、研究記録アプリはサーバーにそのリクエストを送る。
- ⑥ サーバーはリクエストを受け取ると、デバイストークンをデータベースに保管する。

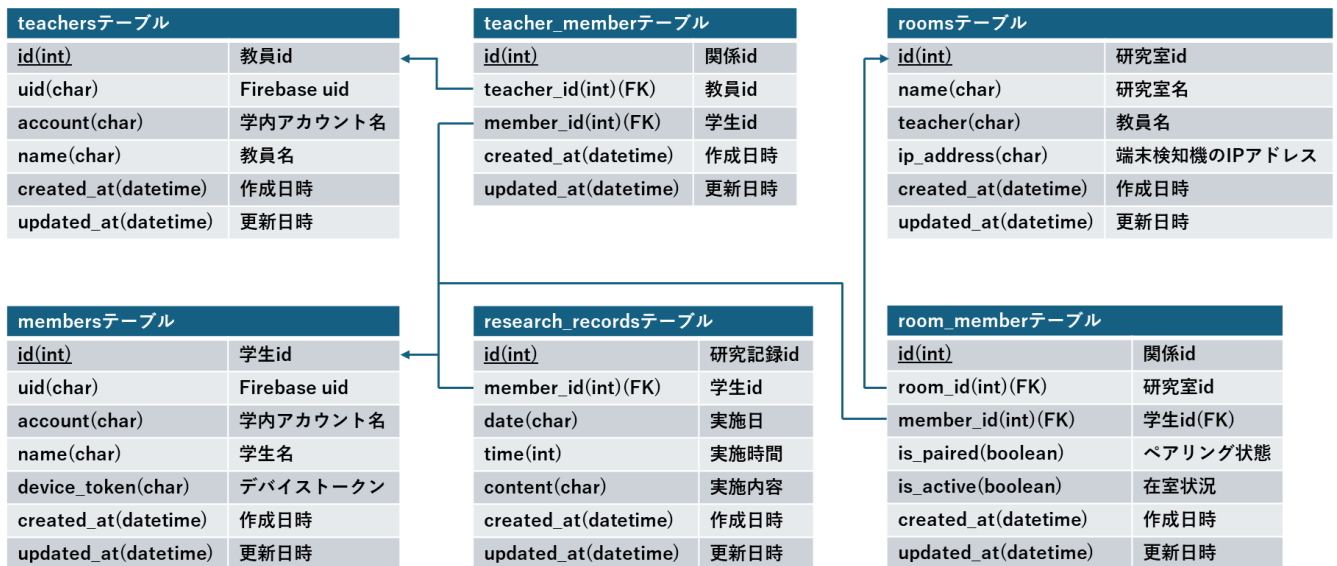


図 5 データベース設計

4.2 データベースの設計

図 5 に本システムのデータベースを示す。各テーブルの id は主キーである。図中の矢印は、各テーブルの外部キーが参照するテーブルを指す。データの管理は関係データベースで行い、関係データベース管理システムは SQLite[6] を使用した。表 1 に各テーブルの役割を示す。

4.3 システムの構成

図 6 に本システムの構成を示す。本システムは主に、端末検知器、サーバー、データベース、FCM から構成されて

表 1 各テーブルの役割

テーブル名	役割
members	学生の情報を管理する。device_token は、FCM が研究記録アプリに割り当てたデバイストークンである。
teachers	教員の情報を管理する。
research_records	学生の研究記録を管理する。研究記録の記録項目である実施日、実施時間、実施内容を各カラムに設定する。member_id で members テーブルを参照する。
rooms	研究室の情報を管理する。ip_address は研究室内の端末検知器の IP アドレスである。IP アドレスは、学生が端末検知器とペアリングするときに使用する。
teacher_member	教員と学生を紐づける。teacher_id で teachers テーブル、member_id で members テーブルを参照する。
room_member	研究室と学生を紐づける。room_id で rooms テーブル、member_id で members テーブルを参照する。is_paired は学生のスマホと端末検知器がペアリングしてある場合、is_active は、学生が研究室に在室している場合に true となる。

いる。端末検知器は研究室内に設置し、端末検知と在室記録を行う。さらに学生が研究室を出入りするたびに在室の有無を示す報告を行い、学生が研究をした後に通知依頼を行う。通知依頼を受けたサーバーは、FCM に通知要求を送る。サーバーは、これ以外に研究記録アプリや教員用機能のページの生成とデータベースの管理をする。研究記録アプリは、学生が学内外問わず研究記録ができるよう、公衆回線からアクセス可能にする必要がある。そこで、サーバーは 443 番ポートのみ開放、FW にて接続できるクライアント IP アドレスを日本国内に限定し、本校 DMZ に配置することでセキュリティ対策を行った。FCM は、サーバーからの通知要求を受け取ると学生に研究実施の記録をスマホに通知する。通知を受け取った学生は場所や時間に関係なく研究記録アプリにアクセスでき、研究記録の記入・閲覧・編集ができる。教員用機能は学内のネットワークからアクセスし、研究室内の学生の在室確認と学生の研究記録の閲覧ができる。本研究では、研究室内の学生の在室確認まで実装した。

4.4 通知から研究記録までの手順

図 6 は、学生が研究室内で研究した後、研究記録アプリで通知を受け取り、研究記録をするまでの流れである。図中の矢印の番号順に説明する。

- (1) 端末検知器は端末検知を行い、学生の在室記録を作成する。
- (2) 学生が研究室を出たことを端末検知器が確認したら、学生の在室記録と FCM への通知依頼をサーバーに送る。通知依頼を送る条件は、一時的な退室や通信不具合を考慮したものとする。本研究では、研究室に 30 分以上在室し、その後、未検知状態が 10 分以上連続した場合とした。
- (3) サーバーは端末検知器から通知依頼を受け取り、通知する学生のデバイストークンをデータベースから取得する。
- (4) サーバーは FCM に通知要求を送る。通知要求には、

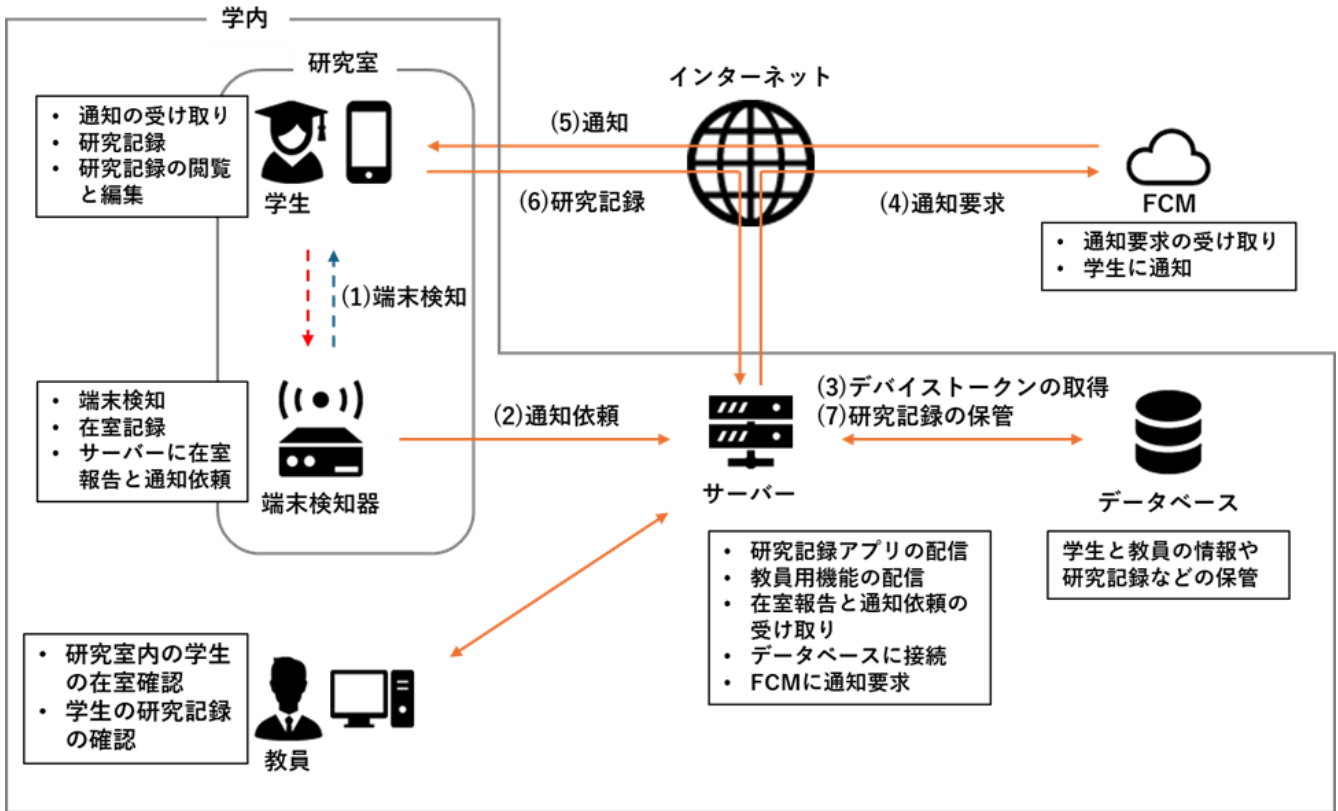


図 6 システム構成

デバイストークンと在室時間を含める。

(5) FCM は通知要求を受け取ると、対象となるデバイストークン宛に通知を送る。

(6) 学生は研究記録アプリで通知を受け取り、研究記録を行う。通知には、研究室の在室時間が表示されているため、研究記録の実施時間の参考にする。ここでの在室時間は、実質的な研究時間ではなく、研究時間の目安とし、後から修正できる。

(7) サーバーは、学生が入力した研究記録を受け取りデータベースに保存する。

5. 開発したシステム

本章では、前章で説明したシステム設計に基づき、開発したシステムの詳細について説明する。

5.1 端末検知器の開発

端末検知器の実行環境を表 2 に示す。端末検知器には Raspberry Pi4 を使用した。Raspberry Pi4 は安価で入手でき、小型であるため設置にコストがかからない。さらに Bluetooth 機能が実装されており、Linux のコマンドラインからそれらを制御できる。ここでは bluetoothctl コマンドと l2ping コマンドを使用する。bluetoothctl コマンドは、Bluetooth の接続状態の確認やペアリングなど制御に関する操作をする。今回はスマホとのペアリングに使用する。l2ping コマンドは、Bluetooth 端末の bd アドレス宛に ping を送る。該当するデバイスからの応答を確認し、端末検知に使用する。全体を統括するプログラムは Python で記述して

表 2 端末検知器の実行環境

項目	名前
実行機器	Raspberry Pi4 model B 4GB
OS	Raspberry Pi OS 64bit
Linux カーネル	6.1.21
Bluetooth	Bluetooth 5.0
言語	Python3.9.2

おり、subprocess ライブラリを使用することで先のコマンドを間接的に呼び出し Bluetooth の各機能を制御する。

5.2 研究記録アプリと教員用機能の開発

研究記録アプリと教員用機能の実行環境を表 3、開発に使用したライブラリを表 4 に示す。サーバーは学内 DMZ 内に設置し、Web サーバーと API サーバーを構築する。Web サーバーは Apache2.4[7]を使用し、研究記録アプリと教員用機能の配信を行う。API サーバーは Python の Web フレームワークである FastAPI[8]を使用し、リクエストの受け取り、データベースとの接続、FCM への通知要求を行う。フロントエンド言語は TypeScript、バックエンド言語は Python を使用する。

まずフロントエンドの開発について説明する。UI 構築は React[9]を使用した。React は JSX 記法に基づいており、リクエスト送信などのロジックとボタンなどの UI を同時に記述できる。画面のデザインには Material UI[10]を使用した。Material UI は、ボタンやテキストボックスなどの UI 部品を用意しており、これらの部品を組み合わせるだけで画面のデザインができる。ログインには Firebase Authentication[11]

を使用する。この Firebase Authentication は、Google が提供する認証機能サービスであり、アプリに簡単にログイン機能を追加できる。なお、熊本高专では Google Workspace 上で割り当てられた学生・教職員アカウントがある。認証機能を利用するシステムで Google Workspace 組織内のアカウントのみにログインを制限でき、これにより、本校の学生と教員のみがログインを許可される。

次にバックエンドの開発について説明する。FastAPI はデータ検証ライブラリの Pydantic[12]と組み合わせて使用する。これにより、リクエストデータの検証が可能になり、フロントエンドとバックエンドでやりとりするデータの整合性が取れるようになる。データベースとの接続には SQLAlchemy[13]を使用した。SQLAlchemy を使用することで、Python からデータベースの SQL 操作ができる。リクエストの認証には、Firebase Admin SDK[14]を使用した。認証は、OAuth2.0 に基づいており、リクエストヘッダに id トークンが含まれているかを検証する。id トークンは、学生と教員が学内 Google アカウントでログインする際に、Firebase Authentication が割り当てる文字列である。

5.3 研究記録アプリの利用

5.3.1 初期登録と端末検知器とのペアリング

図 7 に研究記録アプリで初期登録、および端末検知器とペアリングする際の表示例を示す。学生はスマホの Web ブラウザから研究記録アプリにアクセスし、「ホーム画面に追加」を選択することでホーム画面にアプリを追加する。この操作により、Service Worker はバックグラウンドで動作可能となる。

学生は初回ログイン時に表示される登録画面で通知の許可と研究室の選択を行い、選択した研究室の端末検知器とペアリングする。これらの操作で、アプリは FCM にデバイストークンを要求し、その後、アカウント、FCM デバイストークン、研究室情報などをデータベースに登録する。以降、学生がスマホを所持して研究室を出入りするだけで、端末検知器が自動的に在室を記録し、研究実施の通知を受け取ることができる。

5.3.2 通知の受け取り

図 8 は学生がスマホで通知を受け取った画面例である。研究室を出てから 10 分後に FCM が研究実施の通知を送る。通知はスマホを閉じているときでも受け取ることができる。

表 3 研究記録アプリと教員用機能の実行環境

項目	名前
実行機器	学内 DMZ サーバー
OS	Debian GNU/Linux 12
Linux カーネル	6.1.0
Web サーバー	Apache2.4
API サーバー	FastAPI
フロントエンド言語	TypeScript4.9.5
バックエンド言語	Python3.11.2

表 4 開発に使用したライブラリ

ライブラリ	バージョン
フロントエンド	
React	18.2.0
react-router-dom	6.22.3
Material UI	5.16.4
Firebase	10.11.0
typescript	4.9.5
バックエンド	
FastAPI	0.115.5
uvicorn	0.32.0
SQLAlchemy	2.0.36
SQLAlchemy-Utills	0.41.2
Pydantic	2.9.2
firebase-admin	6.6.0



図 7 初期登録と端末検知器とのペアリング

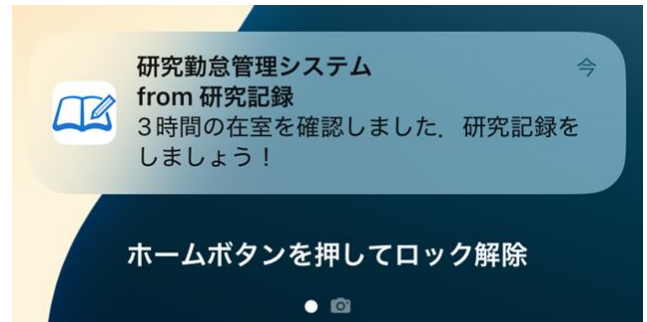


図 8 スマホ通知の受け取り画面例

5.3.3 研究記録と記録内容の閲覧と編集

図 9 に入力画面と編集画面を示す。入力画面では当日の研究記録を行う。記録項目は、実施日、実施時間、実施内容である。実施時間はセレクトメニューで選択し、実施内容はテキストボックスに入力する。実施日は入力ボタンを押したときに自動で取得する。編集画面では月ごとの研究記録の閲覧と編集ができる。タブで月を選択することで、月ごとの研究記録を閲覧できる。メニューボタンを押すと編集ボタンと削除ボタンが表示され、指定の研究記録の編集・削除ができる。追加ボタンを押すと、実施日を直接指



図 10 研究記録の作成手順



図 9 入力画面と編集画面



図 11 教員用機能

定して研究記録を追加できる。これにより、記録忘れの研究や研究室外で実施した研究記録を後から追加できる。ダウンロードボタンを押すと、学校所定の Excel ファイルと研究記録の csv ファイルをまとめた zip ファイルがダウンロードできる。

5.3.4 研究記録の作成

図 10 に研究記録の作成手順を示す。csv ファイルは Excel ファイル内のコピー用シートにコピー&ペーストすることで、Excel 関数が参照し、月ごとのシートに研究記録を自動転記する。出力を csv ファイルとしたのは、今後提出する研究記録のフォーマットが変わった場合や研究実施記録以外の用途に応用する場合、Excel シートや関数を書き直すだけで、出力フォーマットを自由に変更できるためである。

5.4 教員用機能の利用

図 11 に教員用機能を示す。教員は自身の学内 Google ア

カウントでログインする。初回ログイン時に教員情報をデータベースに登録する。テキストボックスに学生名を入力し、追加ボタンを押すことで、学生の在室確認ができる。在室の項目の「○」「×」はリアルタイムな在室状況を反映しており、リロードボタンを押して再取得する。

6. 利用学生からのフィードバック

開発したシステムを同研究室のメンバー4名に使用してもらった。フィードバックは自由記述形式で、本システムを利用して便利に感じた点、本システムを利用して改善してほしい点や気になった点を答えてもらった。

表 5 にフィードバックで得た回答を示す。便利に感じた点については、「記録忘れがなくなった」「研究記録の正確性が上がった」「スマホで研究記録ができる点」などを挙げてくれた。これらの回答から本システムは、研究記録の課題であった記録忘れの防止、および利便性に対して、有効であることがわかった。改善してほしい点や気になった点

については、4人それぞれ異なる回答が得られた。改善点としては、研究室の在室時間の閲覧、ログイン時間の短縮などが挙げられる。また、研究記録も1年のみであり、年度更新など研究記録の引継ぎについては未実装である。

表 5 フィードバック

1. 本システムを利用して便利に感じた点を挙げてください。	<回答者 A> <ul style="list-style-type: none"> ・研究内容を入力する際に前回入力したことがある内容がサジェストされるので手間がかからず便利 ・後から追加や削除などの編集が行える ・年度末に提出しなければならない時にダウンロードがボタンひとつででき、データを移し替える必要もない ・学外からでも入力ができ、スマホで行える ・スマホでの Excel ファイルの編集は操作性にあまりよくなく使いづらい印象だったがこのシステムは操作性がとてもよく使いやすい
	<回答者 B> <ul style="list-style-type: none"> ・アプリの設定がしやすく、研究室を出たらリマインドが来るので記録のつけ忘れがなくなった
	<回答者 C> <ul style="list-style-type: none"> ・スマートフォンを所持しているだけで自動的に在室時間が記録され、研究記録が自動で生成される
	<回答者 D> <ul style="list-style-type: none"> ・研究をする度に通知がくるので、記入を忘れることが減り、活動毎に記録簿に記入することができた ・研究時間を知らせてくれるので、自分がどのくらい活動したかがわかり記録の正確性が上がった
2. 本システムを利用して改善してほしい点や気になった点を挙げてください。	<回答者 A> <ul style="list-style-type: none"> ・実施予定数時数に対してあとどれくらい足りないか分かったら嬉しい ・何時間滞在していたか、通知を後から見返せたら嬉しい ・自分の現在の合計時間について元のフォーマットのような年間の項目があれば嬉しい
	<回答者 B> <ul style="list-style-type: none"> ・サイドバーの「編集」を「研究の記録」にしたほうがいい ・去年の研究記録が閲覧できない
	<回答者 C> <ul style="list-style-type: none"> ・特に不満はないが、ログイン時の待機時間が少し長い
	<回答者 D> <ul style="list-style-type: none"> ・スマホの機種の問題かもしれないが、通知が来ない、ラズパイと接続しない事があった ・研究室を出ないと通知が来ないため、例えば研究室で研究後に他のことをするとき少し面倒

7. まとめ

本研究では、部屋の利用者を自動的に識別・記録する在室管理の方法を提案し、端末検知器を部屋に設置することで、利用者を自動的に識別、利用者の在室を自動的に記録するシステムを構築した。さらに提案した方法を熊本高専の研究活動に応用した。学生用機能として、研究の通知・記録を提供する研究記録アプリを実装し、研究記録忘れの防止と即時入力する環境を提供した。また教員用機能を実装し、研究室内の学生の在室確認機能を提供した。本システムを使用した学生のフィードバックから、本システムが学生の研究記録に対して有効であることを確認した。今後の展望として、システムの完成度を高めるとともに、学生からのフィードバックで得た回答を基にシステムを改良していきたい。

謝辞

本研究を遂行するに際して、熊本高等専門学校 技術職員の岩本舞氏、中村佑介氏、および同じ研究室の城君、石川君、加藤君、嶋中君には、本研究に対するコメントやアドバイスをしていただき、研究を進める上で大きな助けとなりました。ここに感謝を申し上げます。

参考文献

- [1] 東昭太郎, 曾超: Web 在室管理システムにおける無線 LAN を用いた入退室処理の自動化への試み, 情報処理学会第 81 回全国大会講演論文集, Vol. 2019, No. 1, pp. 115-116 (2019).
- [2] 酒井航太, 吉野考, 呉海元: プライバシーを考慮したカードレス在室管理システムの提案, 情報処理学会第 83 回全国大会講演論文集, Vol. 2021, No. 1, pp. 157-158 (2021).
- [3] 増田進也, 小高知宏, 黒岩丈介, 白井治彦: 携帯端末を使用した出席管理システム, 福井大学大学院工学部研究報告, Vol. 65, pp. 37-44 (2017).
- [4] Firebase Cloud Messaging (オンライン), 入手先 <https://firebase.google.com/docs/cloud-messaging?hl=ja> (参照 2025-01-12).
- [5] Sarwar Ali, Chetna Grover, and Renu Chaudhary: Progressive Web Apps(PWAs) – Alternate to Mobile and Web, Emerging Technologies in Data Mining and Information Security Proceedings of IEMIS 2022, Vol. 1, pp. 565-576 (2022).
- [6] SQLite Home Page (オンライン), 入手先 <https://www.sqlite.org/> (参照 2025-01-12).
- [7] Welcome! - The Apache HTTP Server Project (オンライン), 入手先 <https://httpd.apache.org/> (参照 2025-01-19).
- [8] FastAPI (オンライン), 入手先 <https://fastapi.tiangolo.com/ja/> (参照 2025-01-12).
- [9] React (オンライン), 入手先 <https://react.dev/> (参照 2025-01-12).
- [10] Material UI: React components that implement Material Design (オンライン), 入手先 <https://mui.com/material-ui/> (参照 2025-01-12).
- [11] Firebase Authentication (オンライン), 入手先 <https://firebase.google.com/docs/auth?hl=ja> (参照 2025-01-12).
- [12] Welcome to Pydantic – Pydantic (オンライン), 入手先 <https://docs.pydantic.dev/latest/> (参照 2025-01-12).
- [13] SQLAlchemy - The Database Toolkit for Python, (オンライン), 入手先 <https://www.sqlalchemy.org/> (参照 2025-01-12).
- [14] サーバーに Firebase Admin SDK を追加する (オンライン), 入手先 <https://firebase.google.com/docs/admin/setup?hl=ja> (参照 2025-01-12).