

LMS への出席情報記録および連続欠席者提示を行う 出席管理アプリ

橘 弘智¹ 久保田 真一郎² 山場 久昭² 高塚 佳代子² 岡崎 直宣²

概要：教員が個別に管理する出席情報を集約し、学生活動を把握することでエンロールマネジメントに役立てようとする試みがある。記録した学生の出席状況の集計処理と確認は授業終了後に行われるため、連続して欠席している学生の発見が遅れる。本研究では、NFC 搭載 Android 端末で IC 学生証を読み取り LMS に出席情報を記録および集約し、同時に集約された出席情報をもとに連続して欠席している学生をアラート表示するアプリを開発することで、長期欠席の兆候が見られる学生を早期に発見する活動の支援を目指す。開発アプリにより講義で出席をとり LMS へ送信すると即座に、講義を連続して欠席している学生を発見することができ、他の講義の出席状況とあわせて連日欠席している学生を発見することもできる。

キーワード：LMS, IC 学生証, NFC, 連続欠席, エンロールマネジメント

NFC mobile Application for Recording Students Presences on LMS and Alerting Lecturers about Conspicuous Absences

TACHIBANA HIROAKI¹ KUBOTA SHIN-ICHIRO² YAMABA HISAAKI² TAKATSUKA KAYOKO²
OKAZAKI NAONOBU²

Abstract: Some higher education institutions in Japan challenge the enrollment management to figure out student activities based on gathering the information of student presences which lecturers individually hold. A lecturer counts and checks the recorded presences, such as a number of absence or a previous absence. A lecturer tends to be late for noticing the student absences because of checking the record after a class. The goal of our research is to support activities for a lecturer to notice the conspicuous absence of students in a class as early as possible. For early noticing, the developed system has the function to record the student presences and check the conspicuous absences in its same time. So as to record the student presences, we develop the Android application to scan student ID cards equipped with NFC, and record the presence data on a LMS. The application also has a function to check the conspicuous absence as soon as recording the presence on the LMS. Owing to be recorded for the student presences of several classes on one LMS, the application enables a lecturer to check the conspicuous absences across several classes.

Keywords: LMS, IC student ID card, NFC, conspicuous absences, enrollment management

1. はじめに

講義の出席情報を使い学生の活動を把握することでエン

ロールマネジメントに役立てようとする試みがある。学生活動を把握するためには教員が個別に管理する出席情報を集約する必要があり、出席情報を学習管理システム (LMS: Learning Management System) に集約すると管理の面で都合がよい。しかし、既存の出席情報を集約するシステムを使用する場合、出席情報とり LMS に記録するまでに作業コストと作業ミスリスクがあることと、出席情報を LMS に記録する処理や出席情報の集計処理のために別途システ

¹ 宮崎大学工学研究科情報システム工学専攻
Graduate School of Computer Science and System Engineering, University of Miyazaki

² 宮崎大学工学部情報システム工学科
Department of Computer Science and System Engineering, University of Miyazaki

ムを構築する必要があり費用や手間がかかるという問題がある。また、出席情報の集計処理と確認は授業終了後に行われるため連続して欠席している学生の発見が遅れるという問題もある。本研究では、以前開発した、NFCでIC学生証を読み取るAndroidアプリとMoodleのWebサービスAPIを利用することで、IC学生証をAndroid端末で読み取り作成した出席情報を別途サーバ等を使用せずにアプリからMoodleの評定表に直接記録でき、連続欠席者の発見も即座に行えるシステムを提案する。

2. Moodle

Moodleとはオープンソースのeラーニングプラットフォームのひとつであり、フォーラムやクイズ等の機能を持ったオンラインの学習管理システムである。

2.1 Moodleで出席管理することのメリット

Moodleは学習管理システムであるので、学習コンテンツを学生に公開する目的等で教員が自分の開講している講義をコースとして登録している。あわせて学生の情報も登録しているため、新たに学生の名前の情報や講義の情報を入力する必要がなく、出欠を管理するのに都合がよい。また、Moodleには評定機能があり、Moodle上で出題した小テストなどの点数をつけ学生の評価点を記録することができる。加えて、Moodleはオープンソースで開発されており、すでに導入している大学もあるため、その場合は出欠管理システムを別に導入する必要がない。

2.2 WebサービスAPI

MoodleにはWebサービスAPIと呼ばれる、外部のプログラムと相互にメッセージを送受信して連携する機能があり、これを使用することで外部のプログラムにMoodleの機能を組み込むことができる。例えば、外部のプログラムがMoodleに登録されているコースの一覧を取得したり、Moodleへファイルをアップロードしたりといった操作が可能である。MoodleのAPIを呼び出す際には、各機能ごとに決められた形式でMoodleサーバにHTTPリクエストを送信する。そのリクエストを元にMoodleサーバで処理された結果がHTTPレスポンスの形で返ってくる。また、WebサービスAPIの利用には、各ユーザに固有に作成されたトークンが必要で、トークンにより特定のユーザのみがAPIを利用できるように制限される。

3. Moodleに出席情報を集約する既存手法

龍はMoodleが備える出席モジュールを改良し、CSVファイル等で作られた出席データを一括でMoodleに登録できるようにした[1]。しかし、この方法は手入力での出席を入力するため、入力の作業コストや誤入力のリスクが考えられる。

「入力の作業コストや誤入力のリスク」の問題を解決するため伊藤らは教室備え付けのICカードリーダーを利用した出席管理システムを提案している[2]。伊藤らの方法では、講義の際、学生に教室備え付けのICカードリーダーにIC学生証をかざしてもらい出席をとる。そして、とり終わった出席のデータファイルをMoodleにインポートする。この方法ではリーダーでIC学生証を読み取り出席データを作成するため、手入力に比べ手間や誤入力を減らすことができる。しかし、伊藤らのシステムを導入するには、すべての教室にICカードリーダーを備え付けなければならない、導入費用が必要である。

永井らは、Androidタブレット端末とFelicaリーダーを使いIC学生証を読み取る出席管理システムを提案している[3]。永井らのシステムでは、Android端末でとった出席を、事務室等に設置されたアップロード用端末にBluetoothもしくはSMTP経由で送信する。その後、アップロード用端末はLMS連携サーバへ出席データファイルをアップロードし、LMS連携サーバがLMSの出席表を更新する。Android端末は近年低価格化していることに加え、急激に普及していることからすでに所有している教員も少なからずいることが考えられ、伊藤らのように教室備え付けのICカードリーダーを導入するより安価である。しかし、出席データをアップロード用端末にBluetoothで送信する際には、Android端末をアップロード用端末のそばまで持っていく必要がある。また、メールで出席データを送信する場合には、メールの誤送信や未達といったリスクが考えられる。永井らのシステムで利用されるアップロード用端末の構築は、導入時の問題となる。

先行研究において、出席をとる際の省力化のために、NFC搭載Android端末で動作し、IC学生証を読み取るアプリを開発した。このアプリは端末内蔵のNFCを利用するため別途リーダー機器を用意する必要がなく、コストと携帯性に優れる。アプリを動作させた画面を図1に示す。しかし、このアプリで記録された出席データは端末のストレージに保存され、メディアやメール送信により出席データを取り出す必要があった。メール送信には誤送信や未達といったリスクがあり、取り出した出席データファイルをMoodle用に整形するコストと作業ミスリスクが考えられる。

Moodleを使って出席データをとる方法として、「Autoattendance block」[4]というMoodle用のブロックプラグインがあり、このプラグインをMoodleにインストールすることでMoodleに出席を記録する機能を追加できる。この機能では講義開始後に学生がMoodleにログインすることで出席と記録される。Moodle上に自動で出席データが作成されるのでアップロードする必要がない。各学生が自身のPCを講義に持ってきてMoodleにログインするため、各個人のPCを使用する講義はよいが、PCを使用しない講義では利用しやすいといえない。また他人の



図 1 先行研究アプリ動作画面

アカウントでログインし代返するなどの不正も考えられる。

3.1 既存手法の問題点

既存手法の問題点として先に述べたように、出席情報を Moodle に記録する際に作業コストや作業ミスリスクがあることとシステム導入に費用や手間がかかることが挙げられる。加えて、既存のシステムでは出席情報の集計処理と確認は授業終了後に行われるため、長期欠席者になる兆候である連続欠席の発見が遅れ、コミュニケーションをとり出席を促す等の対処ができないせいで長期欠席者を生んでしまうといったことが考えられる。Moodle に記録されている出席情報を自動で集計して連続欠席者に警告のメールを送信するようなシステムも考えられるが、実現には Moodle サーバと出席をとる装置等に加え、出席情報を集計し警告のメールを送信するためのサーバの構築が必要となると考えられる。そこで先行研究で開発した、NFC で IC 学生証を読み取る出席管理アプリと Moodle の Web サービス API を活用することで、別途サーバを必要とせず出席情報の記録、出席情報の取得と集計、集計結果の確認を即座に行えるシステムができるのではないかと考えた。別途サーバにデータベースを用意し出席情報をやり取りするのではなく、Moodle の評定表と直接出席情報をやり取りできれば、システムの構造がシンプルになり作業コストや作業ミスリスクも低くなる。

4. 提案手法

先行研究で開発した NFC で IC 学生証を読み取る Android アプリに Web サービス API で Moodle と出席情報を送受信できる機能を実装した。以後、先行研究で開発したアプリを旧アプリと呼ぶ。本アプリの主な機能として、IC 学生証の読み取り、Moodle との出席情報送受信、連続欠席者のアラート表示があり、連続欠席者のアラート表示機能により既存手法ではできていなかった、長期欠席者の兆候が見られる学生を早期に発見する活動の支援ができる。また、本提案システムでは出席をとる作業や Moodle への出席情報の記録・Moodle からの出席情報の取得をひとつの Android 端末から行えるため、教室備え付けの IC カー

ドリーダや出席記録用のサーバ・端末を使用するシステムと比較して、容易に導入を行うことができる。本アプリの機能詳細と使用の流れを以下に示す。

4.1 IC 学生証の読み取り

IC 学生証の読み取り機能については旧アプリで実装したものである。IC 学生証が Android 端末の NFC 読み取り部にかざされると、かざしたカードの持ち主の氏名、学籍番号をメモリに読み込む。それらのデータとともにカードをかざした時刻を画面に表示するためのリストに追加し、表示している出席表を更新する。同時に出席ファイルとして内部ストレージに保存する。

4.2 Moodle との出席情報送受信

本アプリでは Moodle との出席情報の送受信を Web サービス API により行う。Web サービス API を使う際に必要な HTTP 通信を行うために、非同期 HTTP 通信ライブラリである Android Asynchronous Http Client を組み込んだ。Moodle との出席情報の送受信を実現するための機能として次のものがある。

- Moodle ログイン機能
使用者(教員)が Moodle に登録しているユーザ名とパスワードを使い、アプリにログインする。そのユーザ名とパスワードを Moodle に送信し、トークンを取得する Web サービス API を使うことでトークンを取得する。このトークンは他の Web サービス API を使う際に必要になる。
- Moodle コース取得機能
教員がログインすると同時に、取得したトークンをコース情報を取得する Web サービス API で使い、その教員が登録しているコース名を取得する。取得したコース名をアプリ画面内の講義一覧に追加する。
- 出席情報アップロード機能
学生の IC 学生証のスキャンが終わり、画面上の終了ボタンがタップされると、Moodle の評定表に評定を登録する Web サービスを使い出席情報を記録する。この Web サービス API の利用にも取得したトークンが必要である。記録される形式については後述する。
- 出席情報取得機能
出席情報アップロード機能で評定表に出席情報が登録されると同時に、評定表から評定を取得する Web サービス API で前回の講義までの出席情報を取得する。受講している全学生の出席情報を取得する必要があるため、Moodle のコースに登録されているアカウント情報を取得する Web サービス API を使い、講義科目と



図 2 連続欠席者アラート表示画面

対応するコースに登録されている全学生の学籍番号をあらかじめ取得し利用する。取得した出席情報は後述の連続欠席者のアラート表示で使われる。

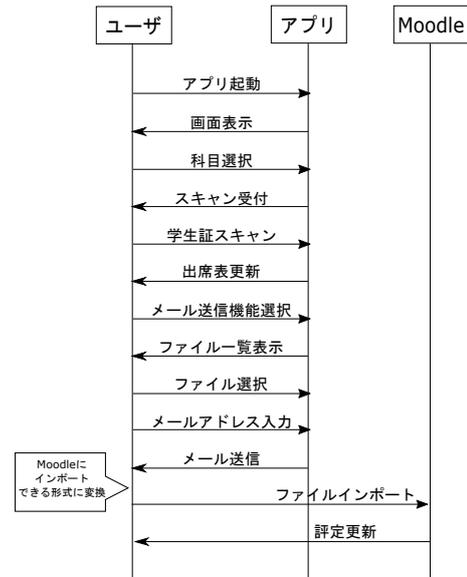


図 3 旧アプリの使用シーケンス

4.3 Moodle での出席情報の処理

開発した Android アプリの出席情報アップロード機能では、指定したコースの評定表に出席状況が登録される。Moodle の評定表は行にコース登録者名、列に評定の項目名が並び、各セルに点数が格納される。そこで今回評定表に出席状況を登録するにあたり、IC 学生証のスキャンした記録をもとにした評定を更新するリクエストをアプリから受け取ると、項目名として、アプリにより追加されたことと講義が開講された日付がわかるように「Scan0526」のように作成し、受け取ったリクエスト内の情報と一致する学生のセルに 1 を格納することで出席を表した。

4.4 連続欠席者のアラート表示

Moodle との出席情報送受信の出席データ取得機能により取得された出席を集計し、連続で 2 回以上欠席している学生の学籍番号、氏名と連続欠席回数が格納されたリストを作成する。作成したリストを Android のアラートダイアログ機能で画面に表示し連続欠席者の情報を使用者(教員)に知らせる。本機能により連続欠席者がアラート表示されたアプリ画面を図 2 に示す。

4.5 アプリ使用の流れ

教員はまず自身の Moodle アカウントでアプリにログインする。ログインが完了すると、自動で読み込まれた Moodle に登録済みの講義リストから出席をとる講義名を選択し、各学生の学生証のスキャンを始める。すべての学生の学生証のスキャンが終わると画面上の読み込み終了ボタンをタップし Moodle に出席情報を送信する。出席情報の送信完了の後、連続欠席者の情報が画面にアラート表示されるので確認する。

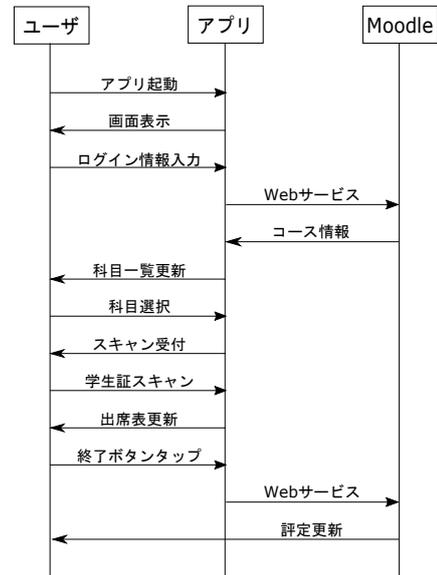


図 4 提案アプリの使用シーケンス

5. 評価

旧アプリと提案アプリで、出席をとる過程から、とった出席情報を Moodle にアップロードし終わるまでにかかった作業経過時間の比較と、Moodle との連携に別途サーバを使用するようなシステムと提案システムで連続欠席者を確認するまでに必要な作業ステップ数の比較により評価を行う。

5.1 作業経過時間の計測

実際に講義で使用し、作業プロセスの経過にかかった時間を計測し比較する。実験での両アプリの作業プロセスは使用の際、毎回行う作業を想定しており、それぞれ図 3 と図 4 のユーザ操作の「アプリ起動」から「評価更新」まで

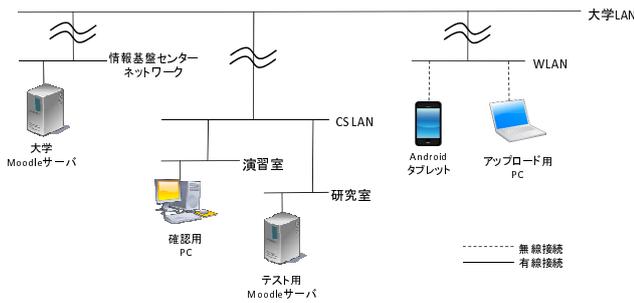


図5 ネットワーク構成

である。評価実験は2015年2月4日13時00分から16時20分まで行われた講義で行った。この講義で学生証を持っている生徒の人数は54人であった。普段から旧アプリで出席をとっている教員1名に両方のアプリを使用して学生証をスキャンしてもらった。旧アプリの場合のアプリからの出席データ取り出しにはメールを使用し、出席ファイルの形式の変換とMoodleへの出席情報のアップロードには、教員のノートPCを使用し、出席情報をアップロードするMoodleには大学が実際に提供しているMoodleを使用した。本論文で提案しているアプリの実験では、テスト用のMoodleサーバを準備し出席情報をアップロードした。この実験で使用したネットワークの構成を図5に示す。2つのアプリを使用して比較するため、講義の始まりに旧アプリ、講義の終わりに提案アプリを使用し2回出席をとり、それぞれ作業プロセスの経過時間を計測した。時間の計測は毎回行う作業を想定し、旧アプリでは「科目選択」から計測を開始し、提案アプリでは「ログイン情報入力」から計測を開始した。また、どちらのアプリの場合も、確認用に用意したデスクトップPCで間違いなく出席が登録できていることを確認した時点で計測終了とした。学生証のスキャンは教員がタブレット端末を持ち、図6の道筋で学生の席を巡回し行った。

5.2 作業経過時間の計測結果

この実験の測定結果を表1に示す。表中の「スキャン時間」は、実験開始からスキャン完了までに要した時間であり、「アップロード時間」は、スキャン完了から計測終了までの時間である。スキャン時間に関しては、提案アプリのほうが42秒長かった。これは提案アプリにのみMoodleへのログイン作業があるためであり、今回の実験ではログイン作業に37秒要したため、スキャンのみにかかった時間には大きな差はなかった。この結果から、提案アプリは旧アプリに比べ合計時間を約6分短縮しており、作業時間を短縮できている。

5.3 連続欠席者確認の即効性についての評価

出席をとるところから、集計された出席情報で連続欠席者を確認するまでの作業ステップの数について、Moodle

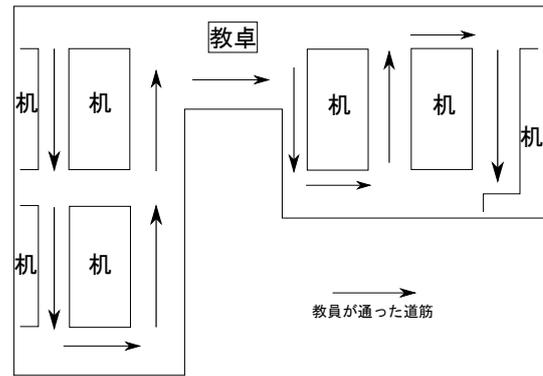


図6 教室全体図

表1 測定結果

	旧アプリ	提案アプリ
スキャン時間	4分4秒	4分46秒
アップロード時間	7分13秒	24秒
合計時間	11分17秒	5分10秒

への出席情報の記録と集計処理を別途用意したサーバ(以下、連携用サーバ)で行うようなシステムと比較する。このシステムでは、Android端末とIC学生証を利用し出席をとり、出席情報を連携用サーバへメールもしくはブラウザ経由で送信した後、連携用サーバがMoodleへ出席情報を記録すると想定する。また出席情報の集計も連携用サーバで行い、教員が集計結果を確認する場合はブラウザで連携用サーバにアクセスし行うこととする。このシステムと提案システムのそれぞれで、出席をとるところから集計された出席情報で連続欠席者を確認するまでの処理シーケンスを図7と図8に示す。図7と図8より、連携用サーバを使用するシステムでは連続欠席者の確認までに6ステップ要するのに対し、提案システムでは5ステップとなっており、連携用サーバを使用するシステムに比べ作業ステップ数が少ない。また、連携用サーバを使用するシステムでは出席情報を送信する際や集計された出席情報を確認する際に、別アプリを起動する必要があるのに対し、提案システムでは出席情報の記録と集計結果の確認をひとつのアプリで即座に行うことができる。

6. 導入時の課題

6.1 Moodle側の準備

本提案システムのアプリからWebサービスAPIを使用しMoodleに出席情報を記録するために、事前に以下の準備をする必要がある。

- モバイル用のWebサービスAPIを有効にする
サイト管理の拡張機能設定画面で「モバイルデバイスのウェブサービスを有効にする」を選択し、モバイル用のWebサービスAPIを使用可能にする。
- 教員用のトークンを発行する
サイト管理のトークン管理画面で「追加」を選択し、

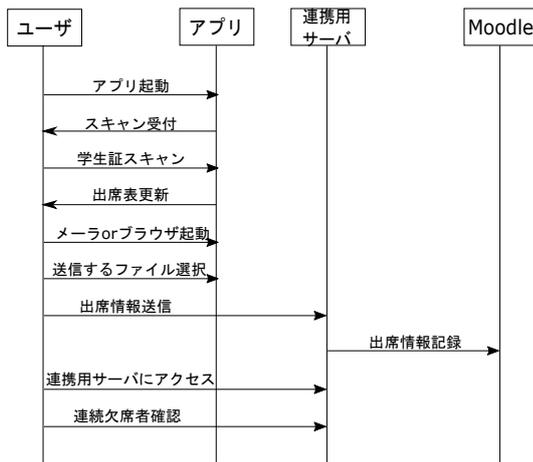


図 7 連携用サーバ使用システム処理シーケンス

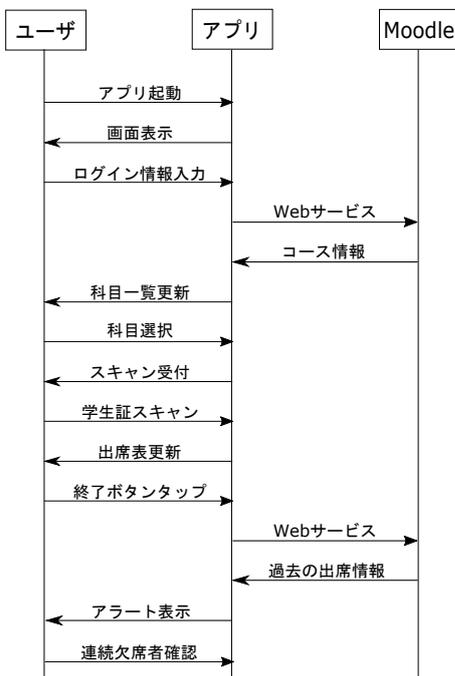


図 8 提案システム処理シーケンス

本提案システムを使用するユーザを選択してトークンを発行する。

- 学籍番号を ID として学生をコースに登録する
本提案システムを使用して出席をとるコースに学生を登録する。この際、学生のアカウントの ID は学籍番号である必要がある。
- ダミーのアクティビティを追加する
評定に記録する出席情報はアクティビティの点数として記録するため、ダミーの小テスト等を作成する。

これらの作業を自動化するには、Moodle プラグインの開発が必要となる。

6.2 他 LMS への対応

本提案システムのアプリは、Moodle 固有の Web サービス

API を使用するため、Moodle 以外の LMS で使用することができないという問題がある。アプリから LMS に出席情報を記録する際の通信を、LMS と学習支援ツールのデータの受け渡しの規格である LTI(Learning Tools Interoperability) に準拠させることで、LTI に対応した異なる種類の LMS で本提案システムが使用できるようになると考えられる。

6.3 他大学学生証への対応

本提案システムのアプリは、宮崎大学の IC 学生証のみを読み取る対象としている。大学ごとに IC 学生証内のデータ保存領域であるブロックの使われ方が違うため、他大学の学生証を正しく読み取ることはできない。Felica 形式の IC カードの共通利用フォーマット (FCF:FeliCa Common-use Format) に準拠している IC カードと IC カードを読み取るシステムでは、互換性があるため、IC カードを使用する団体ごとにシステムに変更を加えることなく使用することができる。本提案システムのアプリを FCF に対応させることで、FCF に準拠した他大学の IC 学生証を使用できるようになると考えられるので、FCF の対応は今後の課題としたい。

7. まとめ

先行研究で開発した NFC で IC 学生証を読み取るアプリと Moodle の Web サービス API を活用することで、IC 学生証でとった出席を別途サーバなしで Moodle の評定表に記録でき、連続欠席者の発見も即座に行えるアプリを開発した。また、先行研究で開発したアプリと提案するアプリの比較、別途サーバを使用し出席情報を Moodle とやり取りするような想定したシステムと提案するアプリの比較により、提案アプリのほうが作業コストや連続欠席者の確認の即効性に関して優れていることを示した。

参考文献

- [1] 龍昌治: LMS を活用した授業実践-Moodle 利用法マニュアル (3) 出席と課題管理編, 愛知大学情報メディアセンター紀要 com 20(1), pp.40-51, (2010)
- [2] 伊藤宏隆, 舟橋健司, 中野智文: 名古屋工業大学における Moodle の構築と運用, メディア教育研究 4(2), pp.15-21, (2008).
- [3] 永井孝幸, 松葉龍一, 久保田真一郎: Android タブレットを用いた FCF キャンパスカード対応 IC カードリーダーのオープンな実装と LMS 連携による出席管理の実現, 学術情報処理研究 (17), pp.67-76, (2013).
- [4] Autoattendance module - MoodleDocs, (online), available from(https://docs.moodle.org/2x/ja/Autoattendance_module), (2015.01.21).