

[研究論文]

# 多機能リマインダの開発 —課題提出支援の検討—

谷村祐<sup>1</sup>・西田滉季<sup>2</sup>・納富一宏<sup>1</sup>

- 1 博士前期課程情報工学専攻  
2 情報工学科

Development of multi-functional reminder system for supporting report submission

Yu TANIMURA<sup>1</sup>, Kouki NISHIDA<sup>2</sup>, Kazuhiro NOTOMI<sup>1</sup>

## Abstract

In this article, we discuss a prototyping of multi-functional reminder to support report submission. We think it is important that students activities on learning at schools are more improved. As the first project, to support to submit reports of courses at school, we developed and evaluated a report submission system on the web site as server program. Our goal is to improve the learning situation at schools, homes and everywhere by the way of e-learning environment also on mobile devices as smartphone and tablet PC. Therefore, as the next project, we propose and implement the prototype of a reminder as Android apps on the smartphones at the client sides. At the client side, Android apps show reminder information to notice. At the server side, personal reminder information are generated, planned and scheduled to the each student. We will evaluate the performance and the effect to the students on real course of our college, and then we will aim to show the results of usability of the system.

Keywords: Educational technology, Reminder, Report submission, Education support and Learning Portfolio

## 1. はじめに

文部科学省の中央教育審議会大学分科会の審議において、学士課程教育の問題点として、学生の学習時間の不足や、成績評価が厳格でないことが指摘されている<sup>1)</sup>。

近年では、学生の主体的な学びを確立し、学士教育課程の質を飛躍的に充実させるために、学士課程教育の質的転換を行うことが喫緊の課題であると示されている。そのために、十分な質を伴った学修時間が増加・確保されているかに着目する必要がある。これは、大学制度において、学修時間は学修の量だけでなく質を伴うものであることが前提であるとされ、また学士教育の質的転換の視点として活用しやすい点や、学士教育の質の保証に不可欠であることなどが理由として挙げられている<sup>2)</sup>。

そこで我々は、質を伴った学修時間を確保する手段として、課題レポートのやり取りに着目した。その理由としては、提出されたレポートにより、質の伴った学修であるかを確認することができるためである。

しかしながら、課題レポートのやり取りを適切に行う際の問題点がある。それは、再提出を繰り返すによる学生の提出意欲低下や、教員の採点や指導の負担増大が挙げられる。この問題は、再提出の回数を減らすことで、対処可能であると考えた。そこで、本研究では、過去の再提出要因の履歴を用いて、提出前に不備の予測と指摘を行う機能を持った課題提出システムの開発を行った<sup>3)</sup>。

この課題提出システムを効果的に運用するためには、課題提出部分を支援する必要がある。そこで、個人の能力を考慮した上で、リマインド情報の提示タイミングや頻度を自動的に設定する機能や、クラスの提出状況を知らせる機能を中心にシステムの試作を行った<sup>4)</sup>。本研究では、予定を忘れることを防ぐために、予定に合わせて通知を行うシステムをリマインダと定義する。また、リマインダによる通知をリマインドと定義する。

本稿では、リマインドを行う Android 端末上で動作するアプリケーションを含めたシステムの詳細について述べる。

## 2. オンラインによる電子的な課題提出

### 2.1 KBook でのオンライン提出

現在本学では数種類の LMS (Learning Management System) が利用されており、課題レポートの電子的な提出ではそれらのシステムを利用していることが多い。ここではこれらのうちの一つである「KBook」の利用について、課題送信数の累積値を Table 1 に、それをグラフに直したものを Fig.1 に示す。

Table 1 課題送信数の累計

	2010 年度	2011 年度	2012 年度
4 月	1,542	987	1,198
5 月	3,131	2,445	2,808
6 月	5,206	4,740	4,774
7 月	7,219	6,539	6,740
8 月	7,228	7,042	6,964
9 月	7,999	9,009	8,228
10 月	9,729	11,887	
11 月	11,122	14,085	
12 月	12,024	15,738	
1 月	12,772	16,844	
2 月	12,844	16,894	
3 月	12,844	16,894	

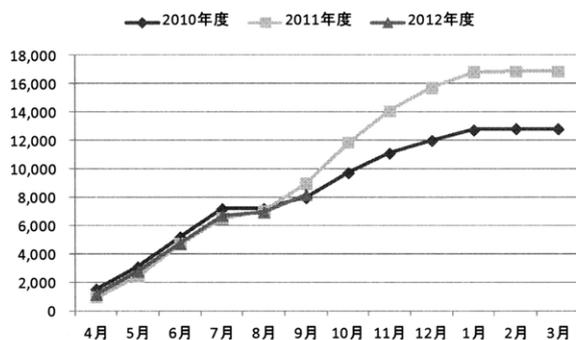


Fig. 1 KBook における課題送信数の累計

このグラフは課題送信数の累計値を月別に表わしている。扱っている課題送信数は情報学部の 3 学科の合計数となっている。累計値は 2010 年度から 2011 年度で増加しており、電子的な方法での課題レポート提出が増えている事がうかがえる。また「KBook」以外の LMS で課題提出を受け付ける、あるいは自作の web ページ、システムで課題提出を受け付けるパターンもあるため、電子的な課題提出はこの数値よりも多く行われている。

### 2.2 課題提出のモデル

本研究で考えている課題提出のモデルを Fig.2 に示す。

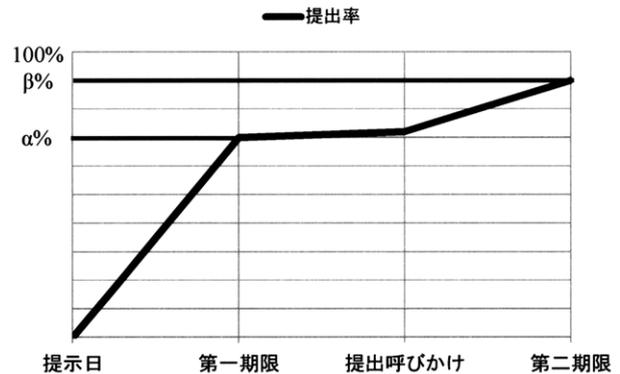


Fig. 2 課題提出のモデル

このモデルは時間の経過による提出率の推移を表しており、横軸が時間、縦軸が提出率を表している。時間は課題を学生に提示する日である提示日と、その提示の際に課題の提出期限として提示される第一期限、第一期限の後に未提出者に再度提出を呼びかける日である提出呼びかけ、最終提出期限である（その後は受け取らない）第二期限の 4 つに区切っている。提出率は、課題を提出した学生数を、履修登録をしている学生数で割った数である。また第一期限時点での課題提出率を  $\alpha$ 、第二期限時点（最終時点）の提出率を  $\beta$  としている。

提出率は図の通り、課題提示日から第一提出期限まで徐々に増えると考えられる。第一提出期限から最終提出期限の呼びかけをするまでは、課題は作成していたが、第一提出期限には間に合わせる事ができなかったものが提出され、少しだけ増えると考えられる。そして、最終提出期限の呼びかけ後から最終提出期限までにある程度提出され、最終的な提出率  $\beta\%$  になると考える。 $\beta$  が 100% ではないのは、提出率の母数が履修登録をしている学生数であり、履修登録をしても単位を修得する意欲が極めて低い（課題を提出する意欲が極めて低い）学生が存在するためである。本学には履修キャンセル期間など途中で履修登録を取り消せる制度が存在しているが、課題提出率が 100% にならない場合があるため、このように考えている。

提出率  $\alpha$ 、 $\beta$  は、単位の重要度、課題の重要度、課題提出の難易度、教員の厳しさ、学生の能力の各要素により決定される。

単位の重要度は、その単位を修得していないと、卒業ができないなど単位を修得しておくことの重要度であり、重要度が高いほど  $\alpha$ 、 $\beta$  ともに高い値になる。特に  $\beta$  には大きく関わってくると考える。これは、単位修得を本気で目指す人数が増えるほど、履修登録はしているが単位を修得する意欲が低い人数が減るためである。

課題の重要度は、単位の修得にどの程度課題の提出が必要とされているかにより変化し、必要性が大きいほど高くなる。この重要度が低いほど、 $\alpha$ 、 $\beta$  ともに低くなると考える。

課題提出の難易度は、課題の量や内容により変わり、

量が多く内容が難しいほど高くなる。この難易度が高いほど、 $\alpha$ 、 $\beta$ ともに低い値になる。これは難易度が高いほど諦めてしまう人が多くなるためである。また $\alpha$ は、期限内に合わない人も増加してしまうため、より影響を受けると考える。

教員の厳しさは、期限をどの程度重要視するかなどで決まる。教員が厳しいほど $\alpha$ は高くなるが、そこから $\beta$ にかけてはあまり上がらないと考える。これは、第一提出期限後に提出を諦めてしまう人が多くなるためである。また厳しくない場合は、 $\alpha$ はあまり高くないが、期限後も諦めずに取り組む人は多くなるため、最終的な $\beta$ の割合はあまり変わらなくなると考える。

学生の能力は、学生の課題作成能力やスケジュール管理能力のことである。能力が高いほど $\alpha$ 、 $\beta$ ともに高くなる。特に $\alpha$ は大きく影響を受けると考える。スケジュール管理能力が高ければ間に合う可能性が高くなるためである。

課題提出率を上げるためには、まず $\alpha$ の値を上げることが重要であると考えられる。これは、期限内に間に合わない学生などはサポートしやすいためである。また、 $\alpha$ を上げることにより教員が第一期限後にサポートする学生数を減らすことができ、より細かくサポートすることができる。これにより $\beta$ の値も上げていけるのではないかと考えている。

### 3. 要件の検討

課題の期限を忘れる、スケジュール管理が不得手であるなどの理由で課題作成が間に合わない学生をサポートするために、事前に通知することで提出期限を思い出させるリマインダが有効であると考えている。

#### 3.1 アンケート目的

リマインダを利用して対処を行う場合、どのような仕様でシステムを作成するべきかを検討するために、アンケート調査を行った。

#### 3.2 アンケート結果

アンケートの対象者は本学学生 20 名とした。アンケートの設問を Table 2 に示し、割合で示すことができる回答を Fig.3 に示す。設問 2「課題の作成・提出をサポートするためにリマインダを使っているか、使おうと思うか」は、リマインダを知っている人はリマインダを利用しているか、知らない人はリマインダの説明を読んだ上でこれからリマインダを利用しようと思うかを確認するための設問である。この設問に対する回答は、知っていて使っているが 0 人、知らなかったが使おうと思うが 5 人となった。このことから、アンケート対象者では、現在利用している人は存在せず、使おうと思う人を合わせても 25%と少ないことが分かった。設問 3「リマインダを使っていない、使おうと思わない理由」の回答は、リマインド時刻などを

設定するのが面倒であるという理由が多数であった。また、使わなくても忘れないという理由もあった。リマインド時刻などを自動で設定してくれるリマインダについては、設問 4「自動で設定を行ってくれるリマインダが便利だと思うか」は yes が 85%、設問 5「自動で設定を行ってくれるリマインダを使いたいと思うか」は yes が 55%となり、自動で設定を行うリマインドが有効である可能性があることを確認できた。設問 7「リマインドされた際にすぐ行動するか」には、90%がすぐには行動しないと思うと回答したが、その理由には、内容によるが急を要すると思えばやるとなど、自分の状況により変わるというものもあった。

Table 2 アンケート設問

番号	設問
1	リマインダを知っているか
2	課題の作成・提出をサポートするためにリマインダを使っているか、使おうと思うか
3	リマインダを使っていない、使おうと思わない理由
4	自動で設定を行ってくれるリマインダは便利だと思うか
5	自動で設定を行ってくれるリマインダを使いたいと思うか
6	5の回答理由
7	リマインドされた際にすぐ行動すると思うか
8	7の回答理由

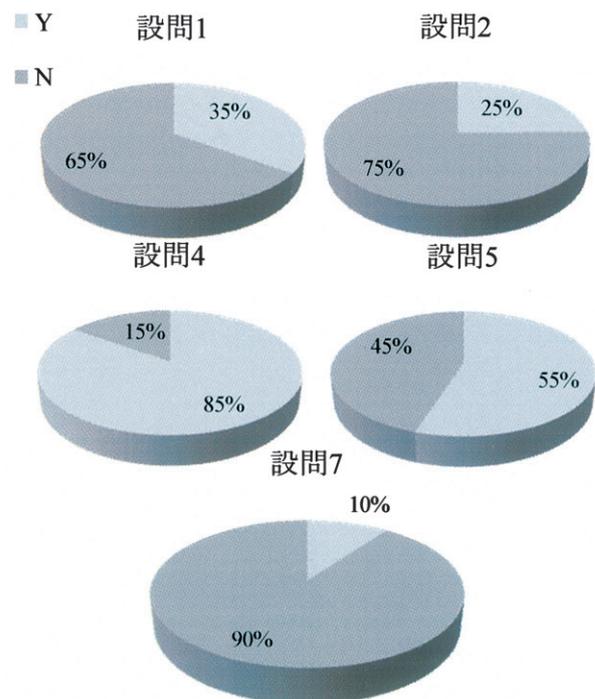


Fig. 3 アンケート結果

### 3.3 要件定義

アンケートの結果から次のことが分かった。

- ① リマインダの設定を行うのは面倒であり、設定してまで使おうとは思わない
- ② リマインドを受けたとしてもすぐに行動する可能性は低い

そこで課題提出を支援するリマインダに必要な要件は、結果①から、リマインド時刻などの設定を自動的に行ってくれること、結果②から、期限に余裕をもってリマインドすることである。また、より効果的にリマインドを行うために、リマインドするタイミングや頻度、内容を個人に合わせるパーソナライズを行うことも必要であるとする。

## 4. システム概要

### 4.1 通知方法

リマインドの通知方法として、メールをするという方法がある。しかし、この方法では、メールをあまり確認しない人には効果が少ないというデメリットがある。そこで、本システムでは、よりリマインドが確認しやすいスマートフォンアプリを用いて、リマインド情報をポップアップで表示する方法をとることにした。

### 4.2 システムの構成

本システムの構成を Fig.4 に示す。本システムは通知用の Android アプリ、サーバ、データベースで構成されており、それぞれ MIT App Inventor, PHP, MySQL を用いて実装した。

MIT App Inventor は、Web ブラウザ上で定義されているブロックを組み合わせて、Android 端末向けのアプリケーションを開発することができるアプリ作成ツールである<sup>5)</sup>。Google によって開発され、現在は MIT(Massachusetts Institute of Technology)で提供されている。

PHP(PHP: Hypertext Preprocessor)は、HTML(Hyper Text Markup Language)に埋め込むことができるスクリプト言語である<sup>6)</sup>。サーバサイドで動作し、動的に Web ページを生成することができる。

MySQL は、RDBMS(Relational DataBase Management System)の 1 つである<sup>7)</sup>。オープンソースであり、データベースの管理を行うことができる。

Android アプリはリマインド通知をポップアップで表示する。サーバはリマインドに関する詳細情報を web 上で表示、リマインドの設定などの処理を行う。データベースは情報の提供を行う。

リマインドの設定を行う際に課題の提出情報を使用する。この情報は課題提出システムが課題を受け取った際に記録される情報を用いる。課題提出システムの構成を Fig.5 に示す。サーバとデータベースはリマインドシステムと同じものを用いている。

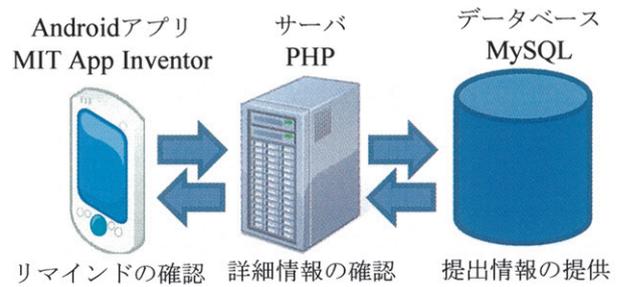


Fig. 4 リマインドシステム構成

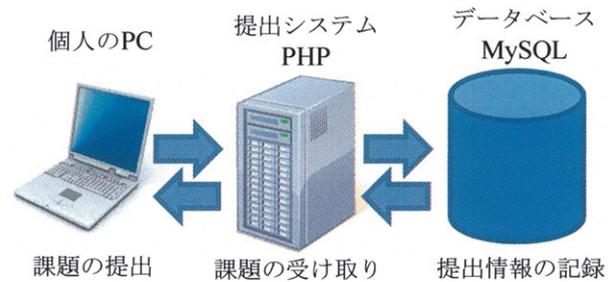


Fig. 5 課題提出システム構成

### 4.3 リマインドのパーソナライズ

本システムでは、ユーザの能力とやる気によりリマインド内容をパーソナライズする方法を提案する。能力は主観的評価(自己評価)と、客観的評価(レポートの採点結果)で判定し、やる気は課題の提出率と課題の提出時刻で判定する。

## 5. システム詳細 (サーバ)

### 5.1 リマインド設定

リマインドの設定は授業ごとに行う。課題を課した際に、PHP のプログラムを実行することで、1 週間分のリマインドタイミング、頻度を纏めて計算し、データベースに記録する。

### 5.2 リマインドタイミング、頻度設定方法

リマインドタイミングはユーザの能力により設定する。能力は、主観的評価と客観的評価をそれぞれ 5 段階で判定する。主観的評価は、自分の能力を 5 段階で自己評価した値を用いる。客観的評価は、過去の課題採点結果を課題ごとに 5 段階で評価し、その平均を評価値として用いる。それぞれの評価値を乗算した結果で、能力を高い・普通・低いの 3 段階に分ける。そして、能力の判定によりそれぞれ提出期限の 1 日前、1・3 日前、1・3・5 日前にリマインドを行うように設定する。

リマインド頻度はユーザのやる気により設定する。課題の提出率と課題の提出時刻をそれぞれ 5 段階で判定する。課題の提出率はそのまま 5 段階で評価を行う。課題の提出時刻は、提出期限に対してどの程度余裕を持って提出

しているかを課題ごとに5段階で評価し、その平均を評価地として用いる。それぞれの評価値を乗算した結果で、やる気がある・普通・ないの3段階に分ける。そして、やる気の判定によりそれぞれ1日に1回・2回・3回リマインドを行うように設定する。タイミングと頻度を設定するとTable 3 のようになり、このような形でリマインドが行われる。

Table 3 リマインド設定

能力/ やる気	ある	普通	ない
高い	提出期限の 1日前に1回	1日前に 2回	1日前に 3回
普通	1,3日前に 1回ずつ	1,3日前に 2回ずつ	1,3日前に 3回ずつ
低い	1,3,5日前に 1回ずつ	1,3,5日前に 2回ずつ	1,3,5日前に 3回ずつ

### 5.3 リマインドの詳細情報

リマインドの詳細情報は web 上で表示し、課題提出期限までの日数、現在までの課題提出率などの詳細なリマインド情報や、ユーザに対するコメント(現在は提出時刻に対するコメントを表示している)、課題ごとのクラスの提出状況のグラフなどの情報を見ることができる。提出状況のグラフは、課題ごとにクラスの提出状況を提出期限に対する提出日別に分け、それぞれの割合を円グラフで表示している。画面を Fig.6 に示す。

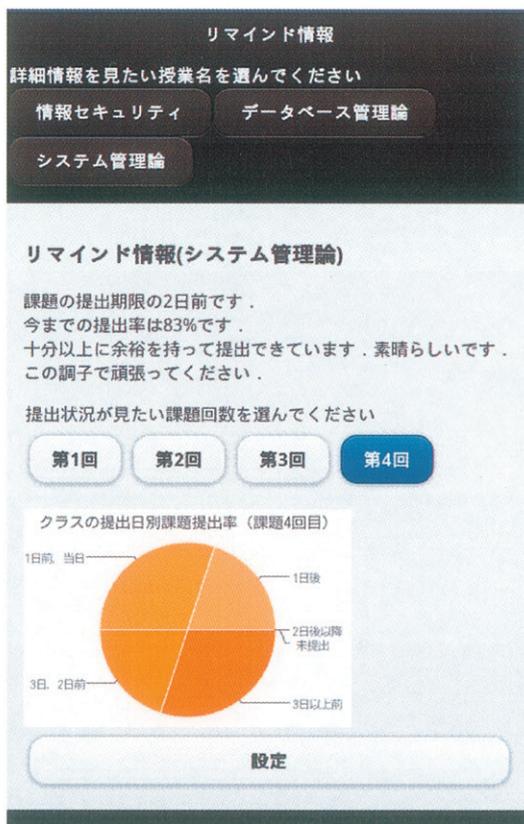


Fig. 6 詳細情報画面

操作手順は、まず詳細情報が見たい授業名をヘッダのメニューから選ぶ。するとボディ部分に選ばれた授業の詳細情報が表示される。その後課題回数を選ぶことで、選んだ回数の課題提出状況を見ることができる。また設定ボタンを押すことで、自己評価ページに移動し、自己評価を設定することができる。

### 5.4 アプリとデータベースの仲介

今回 Android アプリを作成した MIT App Inventor では、課題提出システムで利用している MySQL のデータベースと直接通信することができなかったため、PHP ファイルを介してデータのやり取りを行っている。

具体的には、まず Android アプリが得たいデータに対応した PHP ファイルにアクセスし、PHP ファイルが MySQL に問い合わせを行い、その結果を整形して Android アプリに返すという方法で行っている。

## 6. システム詳細 (Android アプリ)

### 6.1 ID 登録

今回のシステムでは、ユーザ ID として学籍番号を想定している。これは課題提出システムで扱っているユーザ ID が先頭に「s」を付けた学籍番号であるため、同じ形式にしておくことでデータのやり取りを円滑に行えるためである。ID 登録画面を Fig.7 に示す。

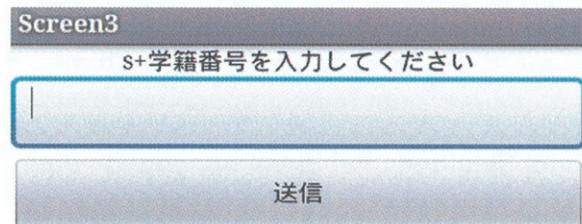


Fig. 7 ID 登録画面

先頭に「s」を付けた学籍番号をテキストボックスに入力し、登録ボタンを押すことでユーザ ID を登録することができる。

この時、サーバに対して入力された値を送信する。値を受け取ったサーバは、データベースに問い合わせを行い、データベースに登録されている学籍番号であるかを確認する。データベースの登録には学籍番号とどの講義を受講しているかをあらかじめ登録しておく。登録してある学籍番号と一致した場合には、その学生が受講している講義の講義名のデータを Android アプリに送信する。登録されている学籍番号と一致しない場合は、空のデータを Android アプリに送信する。

受け取ったデータを Android アプリ側で講義名であるか空であるかを判別し、空だった場合には登録しようとした値が学籍番号では無かったと判断し、登録失敗の通知と値の再入力の要求を行う。講義名だった場合はテキストボックスに入力されたユーザ ID と講義名をそれぞれローカ

ルに保存する。ここで保存したユーザ ID と講義名は、リマインド時刻・内容の確認と設定、リマインド通知の表示の際に利用する。

## 6.2 リマインド時刻設定

リマインド時刻設定は、設定されている値がない場合は、1日に1度リマインド時刻を確認し、リマインド時刻が存在する場合は設定を行う。また、リマインド通知を行った際に、次回のリマインド時刻を取得し設定する。

## 6.3 リマインド通知

リマインド通知は、まず設定されたリマインド時刻に Android アプリが PHP ファイルに問い合わせを行う。そして、問い合わせを受けた PHP ファイルは課題が既に提出されているかどうかを確認する。そこで、提出されていない場合はリマインド内容と次回のリマインド時刻を Android アプリに送信する。しかし、提出されている場合は、処理を終了する。リマインド通知の流れを Fig.8 に示す。

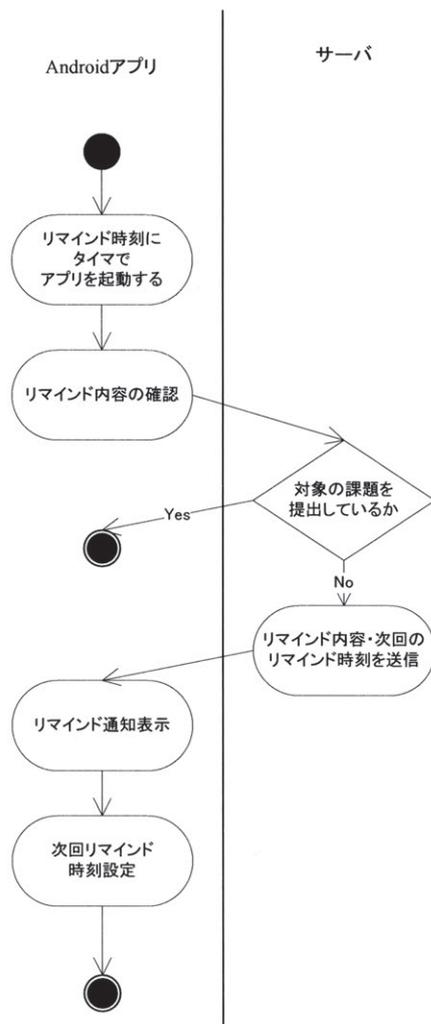


Fig.8 リマインド処理の流れ

Android アプリは、受け取ったリマインド内容をポップアップで表示する。またユーザが画面をみていない場合に備え、端末のバイブレーション機能を利用してリマイン

ド通知があることをユーザに対して通知する。その後、次回のリマインド時刻をタイマにセットする。リマインド通知画面を Fig.9 に示す。

リマインド通知画面の詳細 (Web) ボタンか閉じるボタンのいずれかが押された時に、通知画面は閉じる。たバイブレーションも停止する。そして、詳細(Web)ボタンが押された場合には、Web ブラウザを起動し Fig.6 詳細情報ページにアクセスする。

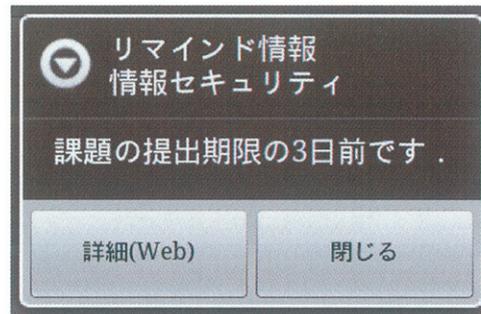


Fig.9 リマインド画面

## 7. おわりに

課題提出のサポートを行うことを目的に、課題提出モデルの検討やリマインダの調査を行った。それらの結果から、個人の能力を考慮し、リマインド情報の提示タイミングや頻度を自動的に設定する機能、クラスの提出状況を知らせる機能、リマインドを行う Android アプリを持つリマインドシステムを開発した。

今後、利用した際の提出率の変化により評価することや、パーソナライズの改良が必要である。

## 参考文献

- [1] 学士課程教育の構築に向けて (答申), [http://www.mext.go.jp/component/b\\_menu/shingi/toushin/\\_icsFiles/afieldfile/2008/12/26/1217067\\_001.pdf](http://www.mext.go.jp/component/b_menu/shingi/toushin/_icsFiles/afieldfile/2008/12/26/1217067_001.pdf) (2008).
- [2] 予測困難な時代において生涯学び続け、主体的に考える力を育成する大学へ (審議まとめ), [http://www.mext.go.jp/component/b\\_menu/shingi/toushin/\\_icsFiles/afieldfile/2012/04/02/1319185\\_1.pdf](http://www.mext.go.jp/component/b_menu/shingi/toushin/_icsFiles/afieldfile/2012/04/02/1319185_1.pdf) (2012)
- [3] 谷村祐, 納富一宏: レポート作成支援機能付き課題提出システムの開発, 情報処理学会 第74回全国大会講演論文集 第4分冊 6ZH-6, pp.959-960, (2012)
- [4] 谷村祐, 納富一宏: 課題提出を支援する多機能リマインダの検討と試作, ヒューマンインタフェース学会 ヒューマンインタフェースシンポジウム 2012, 1424S, pp.207-208, (2012).
- [5] MIT App Inventor, <http://appinventor.mit.edu/>
- [6] PHP: Hypertext Preprocessor, <http://php.net/>
- [7] MySQL, <http://www-jp.mysql.com/>