

2011 年度 渡辺テーマ研究会

「旅行に関する情報行動の効率化と支援機構の検討」

担当教員 渡辺 恭人  
学籍番号 0840213  
武田 竜介

提出日：2012 年 1 月 25 日

## 目次

1. 背景・目的.....	4
1-1. 背景.....	4
1-2. 目的.....	4
2. 現状と問題点.....	5
2-1. 現状.....	5
2-2. 既存のサービス.....	6
2-2-1. 宿泊予約サイト.....	9
2-2-1-1. じゃらん net.....	9
2-2-1-2. 楽天トラベル.....	11
2-2-1-3. JTB.....	14
2-2-2. 乗換案内、ルート検索.....	16
2-2-2-1. ジョルダン.....	16
2-2-2-2. 駅探.....	17
2-2-2-3. NAVITIME.....	18
2-2-2-4. Goo 地図.....	19
2-2-2-5. Google map.....	20
2-2-3. 天気予報.....	21
2-2-3-1. 気象庁.....	21
2-2-3-2. Yahoo! 天気情報.....	21
2-2-3-3. weathernews.....	22
2-2-4. 飲食店検索.....	23
2-2-4-1. ぐるなび.....	23
2-2-4-2. ホットペッパーグルメ.....	24
2-2-4-3. 食べログ.....	25
2-3. 現在の問題点.....	26
3. 解決方法の検討.....	28
3-1. 使用サイトの検討.....	28
2-3-1. 宿泊予約サイト.....	28
2-3-2. ルート検索.....	28
2-3-3. 天気予報.....	29
2-3-4. 飲食店検索.....	29
3-2. 解決に対するアプローチ.....	29
4. 構成機能・要素.....	31
4-1. 構成機能.....	31

4-2. 構成要素.....	31
5. 設計	
5-1. 実装環境.....	33
5-2. データの取得.....	33
5-2-1. ウェブアプリからネイティブアプリへの変換.....	33
5-2-2. 宿泊施設予約機能の設計.....	37
6. 動作の検証と評価.....	48
6-1. 動作環境.....	48
6-2. 動作画面.....	48
6-3. 機能評価.....	50
6-4. 考察.....	51
7. まとめと今後の課題.....	52
7-1. まとめ.....	52
7-2. 今後の課題.....	52
参考文献.....	53
謝辞.....	55

## 1. 背景・目的

### 1-1 背景

旅行には、行く前、現地、帰った後のそれぞれに多くの情報が必要になります。

旅行に行く際に携帯電話を利用して、交通手段や食事処の検索を利用する人が多くいます。

旅行の際に自分の現在地や向かっている場所など条件を変えるたびに検索しなおさなくてはなりません。

楽しんでいる時間を割き、繰り返し入力するのは煩雑で非効率です。

この時間を短くし、旅行を有意義に過ごすために現在使用されているウェブサイトを利用し、旅行を有意義に過ごすためのスマートフォン等の携帯端末を使用する方法を検討する。

### 1-2 目的

研究目的は、旅行に行く際、旅行の最中など情報行動の手間を簡略化し、旅行をより充実したものにする事です。

現存する、複数の情報を統括することで複数のウェブサイトをもたぎ、再入力の手間をなくします。

この中から最も使いやすい物もしくは情報量の多い物を選択し利用します。

## 2. 現状と問題点

### 2-1 現状

旅行には人それぞれに旅行の目的があり、旅行時にする行動は異なってきます。  
旅行の目的は多岐にわたります。

- ・ 疲れを癒しに温泉に入る
- ・ 素晴らしい旅館を体験
- ・ その土地でしか味わえない郷土料理を味わう
- ・ その土地で行われる祭に参加
- ・ スポーツを観戦
- ・ スポーツを体験
- ・ 旅行までに行程を楽しむ
- ・ 景色を楽しむ
- ・ 伝統芸能を体感
- ・ 世界遺産、文化遺産を見る
- ・ 美術館、博物館で知的欲求を満たす
- ・ テーマパークで遊ぶ
- ・ 自然を味わう
- ・ 健康の為に体を動かす

などのように人それぞれに旅行の目的があり、複数の旅行目的がある人もいます。  
旅行時に行う情報行動としても

- ・ 旅行に行く前には宿泊先の予約を予約サイト
- ・ 旅行先での有名な場所や話題の場所などを検索
- ・ 旅行前日には旅行先の天気を天気予報
- ・ 旅行先までを電車で移動なら路線検索
- ・ 車やバイクならルート検索
- ・ 旅行先で食べたいものを探すために飲食店検索
- ・ 旅行先から戻るのにまた路線検索やルート検索
- ・ 旅行から戻って来たら、写真の整理や日記などの思い出の整理

などの多くの情報行動が予測されます。

## 2-2 既存のサービス

旅行時に利用する情報行動は旅行の目的によって使い分けられます。

宿泊予約サイト、交通網の検索、飲食店の検索、天気予報などの情報行動も現在では多くのサイトが存在します。

ここでは、情報行動でどのようなものが使われているか、その情報行動はどのようなサイトが存在するかを分析します。

表 2-2 旅行に利用される情報行動の一覧

	内容	情報	手段
行く前 準備	<ul style="list-style-type: none"> <li>・ 旅行内容の下調べ</li> <li>・ 宿泊場所の予約</li> <li>・ 移動手段の予約</li> <li>・ 旅行地周辺検索</li>   <li>・ 天気の確認</li> <li>・ 旅行目的の予約</li> <li>・ ルートの確認</li>   <li>・ SNS などのつぶやき</li> </ul>	<ul style="list-style-type: none"> <li>・ 旅行目的となる情報</li> <li>・ 宿泊施設情報</li> <li>・ チケット情報</li> <li>・ 旅行目的地周辺情報</li>   <li>・ 天気情報</li> <li>・ 旅行目的の予約</li> <li>・ 地図情報</li>   <li>・ 情報の共有</li> </ul>	<ul style="list-style-type: none"> <li>・ 検索サイト</li> <li>・ 宿泊予約サイト</li> <li>・ 旅券予約サイト</li> <li>・ 検索サイト</li> <li>・ 地図サイト</li> <li>・ 天気予報サイト</li> <li>・ 各種予約サイト</li> <li>・ ルート検索サイト</li> <li>・ 地図サイト</li> <li>・ SNS サイト</li> </ul>
行くとき	<ul style="list-style-type: none"> <li>・ ルート検索</li> <li>・ 電車の時刻検索</li> <li>・ 飲食店検索</li> <li>・ SNS などのつぶやき</li> <li>・ 写真の撮影</li> <li>・ 旅行目的の確認</li> <li>・</li> </ul>	<ul style="list-style-type: none"> <li>・ 地図情報</li> <li>・ 時刻表情報</li> <li>・ 飲食店情報</li> <li>・ 情報の共有</li> <li>・ 情報の記録</li> <li>・ 旅行目的の情報</li> <li>・</li> </ul>	<ul style="list-style-type: none"> <li>・ ルート検索サイト</li> <li>・ 時刻表検索サイト</li> <li>・ 飲食店検索サイト</li> <li>・ SNS サイト</li> <li>・ カメラ</li> <li>・ 各種予約サイト</li> <li>・ 検索サイト</li> <li>・</li> </ul>
現地	<ul style="list-style-type: none"> <li>・ ルート検索</li> <li>・ 電車などの時刻検索</li> <li>・ 飲食店検索</li> <li>・ SNS などのつぶやき</li> </ul>	<ul style="list-style-type: none"> <li>・ ルート情報</li> <li>・ 時刻表情報</li> <li>・ 飲食店情報</li> <li>・ 情報の共有</li> <li>・ 情報の記録</li> <li>・ 旅行目的の情報</li> </ul>	<ul style="list-style-type: none"> <li>・ ルート検索サイト</li> <li>・ 時刻表検索サイト</li> <li>・ 飲食店検索サイト</li> </ul>

	<ul style="list-style-type: none"> <li>写真の撮影</li> <li>旅行目的の確認</li> </ul>		<ul style="list-style-type: none"> <li>SNS サイト</li> <li>カメラ</li> <li>各種予約サイト</li> </ul>
帰るとき	<ul style="list-style-type: none"> <li>ルート検索</li> <li>電車の時刻表検索</li> <li>飲食店検索</li> <li>SNS などのつぶやき</li> <li>写真の撮影</li> </ul>	<ul style="list-style-type: none"> <li>ルート情報</li> <li>時刻表情報</li> <li>飲食店情報</li> <li>情報の共有</li> <li>情報の記録</li> </ul>	<ul style="list-style-type: none"> <li>ルート検索サイト</li> <li>時刻表検索サイト</li> <li>飲食店検索サイト</li> <li>SNS サイト</li> <li>カメラ</li> </ul>
帰った後	<ul style="list-style-type: none"> <li>写真の整理</li> <li>ブログなどの更新</li> <li>SNS などのつぶやき</li> </ul>	<ul style="list-style-type: none"> <li>写真の整理</li> <li>ブログなどの更新</li> </ul>	<ul style="list-style-type: none"> <li>画像編集</li> <li>ブログサイト</li> <li>SNS サイト</li> </ul>

旅行場所を決める場合、目的地が決まっているか旅行の内容が決まっている場合が多いです。

どちらの場合でも、目的地周辺の情報や旅行目的に適した場所を知るために検索サイトを利用します。

目的地が決まると、目的地や旅行目的に適した宿やローケーションの良い宿を予約するために宿泊予約サイトを利用します。

目的地までの道程や旅行中、帰り道に空腹を満たすために自分の食べたい物を探す目的で飲食店検索サイトを利用します。

また、旅行中や旅行後に自分の気になったものや感動したもの、心動いたものを他人と共有するために SNS サイトを利用します。

その中でも、旅行時に使用される頻度の高い宿泊の予約、交通網の検索、飲食店の検索これに加えて、日常生活から利用頻度の高い天気予報を主に分析します。



## 2-2-1 宿泊予約サイト

旅行時の宿泊する施設を検索、予約するためのサイト

施設の設備情報や電話番号、住所などが掲載されている。

### 2-2-1-1 じゃらん net

じゃらん net(図 2-2-1-1(1))は登録件数が 20,922 軒、宿泊場所を検索するために、「エリア、宿泊日数・部屋数、合計人数、予算」から検索することができます。

その他に、PC のサイトを利用することによって、「今晚の宿泊予約、出張ビジネス、温泉ガイド、高級宿、日帰り情報、航空券+宿、ツアーレンタカー、高速バス、航空券、観光ガイド、海外」を検索することが可能です。

宿の情報には「料金、クチコミ、施設詳細、住所、交通アクセス、パーキング、部屋の種類、部屋の説明補足、部屋の設備情報、施設内の温泉・風呂情報、アメニティ、施設内容、サービス、使用可能クレジットカード、チェックイン・チェックアウト時間、旅館補足、プラン一覧」があり、住所には MAP 表示があり、iPhone 内のマップと連動しています。

じゃらん net では、宿の写真がトップの一枚しか掲載されていないため、宿の一部しか確認することができず、内装や風呂などがどうなっているのか確認することができない。

また、スマートフォンからの場合レンタカーやバスの手配のために PC 用のサイトに移動しなくてはならない。

使用可能 API

- ・宿表示
- ・空室検索
- ・エリア検索
- ・温泉検索

使用方法

検索方法は 2 種類あり、キーワードを入力して検索する方法と日付、エリアを選択して検索する方法があります。

選択して検索する場合、どの県のどの地域に行きたいか(例:県/神奈川県、エリア/箱根)宿泊日数と部屋の数、合計で宿泊する人数、予算の下限と上限の4種類を入力することでその条件にあった宿泊施設が表示されます。



図 2-2-1-1(1) じゃらん net トップページ出典【1】



図 2-2-1-1(2) じゃらん net 神奈川県箱根を条件にした検索結果



図 2-2-1-1(3) じゃらん net 箱根 雪月花の施設情報

## 2-2-1-2 楽天トラベル

楽天トラベル(図 2-2-1-2(1))は登録宿泊施設数が国内 24,205 軒、海外 15,459 軒と今回調べた宿泊登録サイトでは最も多く、宿泊場所を検索するために、「行き先、今晚の宿、駅・空港近く、高速バス、航空券+宿、キーワード」が使用可能であります。

知ることのできる特集・キャンペーン情報には「ホテル・宿一覧、日帰り・デイクース、駅・空港付近の宿、連休空室検索、高級ホテル・旅館、ホテルチェーン、東京ディズニーリゾート、ユニバーサル・スタジオ・ジャパン、観光案内たびノート、温泉地情報」があります。

その他の利用可能サービスには、「楽パック(交通+宿泊)、高速バス、レンタカー、海外航空券+ホテル、海外航空券、海外ホテル」があります。

宿の情報には、「空室検索、プラン、日帰り情報、口コミ、フォトギャラリー、施設のおすすめ情報、アクセス、チェックイン・アウト、住所、駐車場、部屋設備、館内設備、特典、マイレージ、食事場所、カード、条件注意事項、キャンセルポリシー、徴収所のポリシー、その他」の情報がありすべてウェブサイトへのリンクで表示されます。

楽天トラベルでは、map 情報が掲載されていないため住所をメモするか、コピーするかして別の map アプリやルート検索サイトを利用する必要があります。

#### 使用可能 API

- ・施設検索
- ・施設情報
- ・地区コード
- ・キーワード
- ・ホテルチェーン

#### 使用方法

検索方法には、キーワードを入力して探す方法、日付、行き先、今晚の宿の 4 つの方法があります。

日付はチェックイン、チェックアウト、利用人数、部屋数、料金の下限、上限を入力して検索をすると、地域を選択画面が表示されるので目的地を選ぶことによって、条件の合った宿泊施設が表示される。

行き先はエリアが表示されるので、その中から目的地を選択することで、その地域にある宿泊施設が表示される。

今晚の宿は、地域を選ぶことで選択した中で今晚泊まれる宿泊施設を表示する。



図 2-2-1-2(1) 楽天トラベル トップページ出典【2】



図 2-2-1-2(2) 楽天トラベル 神奈川県箱根を条件にした検索結果



図 2-2-1-2(3) 楽天トラベル 箱根 雪月花の施設情報

### 2-2-1-3 JTB

JTB(図 2-2-1-3(1))は登録宿泊施設数がわかりません。

宿を検索するために、「旅館・ホテル、国内ツアー、海外ツアー、レジャーチケット」を利用することができます。

JTB のおすすめがあり、「温泉特集、特別プライスで泊まれる宿、学生旅行」などがあり、学生旅行には国内外の情報があります。

JTB のおすすめは PC 専用サイトに移動します。

宿情報には「客室、フォトギャラリー、アクセス、交通・送迎、周辺施設情報」が掲載されています。

他のサイトに比べて宿の情報が少ないのと、利用可能な API が少ない。

利用可能 API

- ・施設検索

使用方法

検索方法はキーワードを入力して検索する方法と条件を入力して検索する方法があります。

条件から探すは宿泊開始日、何泊するか、人数、部屋数、目的地を入力することに条件にあった宿泊施設が表示される。



図 2-2-1-3(1) JTB トップページ出典【3】



図 2-2-1-3 (2) JTB 神奈川県箱根を条件にした検索結果



図 2-2-1-3 (3) JTB 箱根 雪月花の施設情報

## 2-2-2 乗換案内、ルート検索

### 2-2-2-1 ジョルダン

ジョルダン(図 2-2-2-1(1))は検索に「乗換案内、時刻表、地図、運行情報」が使用可能ですが、無料で使用できる機能はごくわずかで有料登録をすることにより様々な情報で検索することが可能です。

API は有料の法人向けのみです。



図 2-2-2-1(1) ジョルダン トップページ出典【4】



図 2-2-2-1(2) ジョルダン 条件の入力画面

### 2-2-2-2 駅探



駅探(図 2-2-2-2(1))は検索に「乗換案内、電車時刻表、運行情報、新幹線時刻表、運行情報、駅情報、路線図」が使用可能で、有料登録により、その他にも情報の検索が可能になります。

API は有料法人向けのみです。



図 2-2-2-2(1) 駅探 トップページ出典【5】



図 2-2-2-2(2) 駅探 条件の入力画面

### 2-2-2-3 NAVITIME

NAVITIME(図 2-2-2-3(1))検索するために、「スポット検索、現在地、乗換案内、時刻表、混雑レポート」が可能です。

API は有料法人向けのみです。



図 2-2-2-3(1) NAVITIME トップページ出典【6】



図 2-2-2-3(2) NAVITIME 条件の入力画面

#### 2-2-2-4 Goo 地図

Goo 地図(図 2-2-2-4(1))はスマートフォン専用サイトがありません。

検索以外に「ルートガイド、地図検索、古地図、注目スポット、統計地図」などがあります。

API はないようです。



図 2-2-2-4(1) Goo 地図 トップページ出典【7】



図 2-2-2-4(2) Goo 地図 条件の入力画面

## 2-2-2-5 Google map

Google map(図 2-2-2-4(1))は「ルートガイド、航空写真、地形、交通状況」が検索可能です。

API は地図、ルート検索など複数のものが利用できます。



図 2-2-2-4(1) Google map トップページ出典【8】



図 2-2-2-4(2) Google map 条件の入力画面

### 2-2-3 天気予報

### 2-2-3-1 気象庁

気象庁(図 2-2-3-1(1))は表示情報に、「天気予報、気象警報、レーダー、地震、台風、津波」などがあります。

API は有料のもののみです。



図 2-2-3-1(1) 気象庁 トップページ出典【9】

### 2-2-3-2 Yahoo!天気情報

Yahoo!天気情報(図 2-2-3-2(1))は表示情報に「天気予報、気象警報、地震、津波、台風、気象レーダー、衛星画像、天気図」があります。

API は海外版のみあります。



図 2-2-3-2(1) Yahoo!天気情報 トップページ出典【10】

### 2-2-3-3 weathernews

weathernews(図 2-2-3-3(1))は表示情報に、「天気予報、気象警報、地震、津波、台風、雨雲レーダー、衛星画像、天気図」があります。

API はありません。



図 2-2-3-3(1) weathernews トップページ出典【11】

### 2-2-4 飲食店検索

## 2-2-4-1 ぐるなび

ぐるなび(図 2-2-4-1(1))は登録店舗数が約 76,000 店舗で、店舗の記載情報には「店舗紹介、メニュー、アクセス、基本情報、口コミ」があります。

その他に、「オススメ、エリア、料理、目的・シーン」で検索が可能です。

使用可能 API

- ・ 飲食店情報
- ・ 業種情報
- ・ 地域情報



図 2-2-4-1(1) ぐるなび トップページ出典【12】



## 図 2-2-4-1(2) ぐるなび 千葉商科大学周辺の検索結果

### 2-2-4-2 ホットペッパーグルメ

ホットペッパーグルメ(図 2-2-4-2(1))は登録店舗数が約 570,000 店舗で記載情報には「店舗紹介、メニュー、アクセス、詳細情報」があります。

その他に「特別企画、テーマ・シーン別グルメ特集」などの特集があります。

使用可能 API

- ・グルメサーチ
- ・店名サーチ
- ・マスタ API 群



図 2-2-4-2(1) ホットペッパーグルメ トップページ出典【13】





図 2-2-4-2(2) ホットペッパーグルメ 千葉商科大学周辺の検索結果

### 2-2-4-3 食べログ

食べログ(図 2-2-4-3(1))は登録店舗数が 655,905 店舗、記載情報には「メニュー、口コミ、写真、詳細情報」があります。

特集にベストレストランが記載されます。

使用可能 API

- ・ レストラン情報
- ・ レストラン口コミ
- ・ レストラン画像



図 2-2-4-3(1) 食べログ トップページ出典【14】



図 2-2-4-3(2) 食べログ 千葉商科大学周辺の検索結果

### 2-3 現在の問題点

旅行時に利用する情報行動が多く、それぞれその場で利用する情報行動を調べなくてはならない。

利用するものが1つならばよいが多くの場合複数の情報行動を利用するため、複数のサイトやアプリを利用しなくてはならない。

これをなるべく無駄が少なくなるように1つのアプリ内で利用できるようにするにはどうすればよいか。

今回情報行動の内容を調べてわかったがAPIを有料や公開していないものが多く、数少ないものの中から何を基準に利用するものを選択するか。

利用者の使いやすいものは当然だが、情報の種類や情報の量が関わってきます。

中でも宿泊予約サイトと飲食店検索は情報が少なくでは利用価値も下がってしまいます。

使用したときにできるかぎり選択しは多い方が良いと考えています。

### 3. 解決方法の検討

旅行時に利用する情報行動のうち利用頻度の高いものを選び、その中から最も適していると思われるものを選択します。

#### 3-1 使用サイトの検討

使用するサイトは第2章で調べたものの中から選択する。

##### 3-1-1 宿泊予約サイト

サイト名	じゃらん net	楽天トラベル	JTB
比較内容			
掲載数	20,922 軒	24,205 軒	不明
利用可能 API	4 種類	5 種類	1 種類
地図	iPhone のマップと連動	MAP なし	AlpsMAP 使用

宿泊予約サイトでは、今回調べた「じゃらん net」、「楽天トラベル」、「JTB」の中で「JTB」では利用できる API が 1 つしかなく、情報を提供するには適していないと判断します。

「じゃらん net」、「楽天トラベル」は宿泊場所を検索という観点のみで判断すると検索可能な情報に差はほとんどないため、記載宿泊情報と利用可能な API の多い「楽天トラベル」が最も希望を満たしているので利用することにする。

##### 3-1-2 ルート検索

サイト名	NAVITIME	Goo 地図	Google map
比較内容			
API	有料	なし	ほとんどの機能が使用可能
検索可能なルート	徒歩、車、電車、ただし一部有料	徒歩、車、電車	徒歩、車、電車 AlpsMAP 使用
スマートフォン専用サイト	有り	無し	有り

の有無			
-----	--	--	--

ジョルダン、駅探もありますが検索可能なものが電車での交通情報のみだったので、排除しました。

ルート検索では、今回調べたもののほとんどが API を有料もしくは公開していなかったため、ルート検索では唯一公開していた「Google map」を利用することにする。

ルート検索では一部は乗換案内のみ所もあり、「Google map」では、歩き、車、電車と主に使われる交通手段の検索もできるため一番適している。

### 3-1-3 天気予報

天気予報は今回調べたものには、API を無償提供しているサイトがないため他の無償で利用可能なものを探した結果 Google が非公式ながら Google weather を提供しているのをそれを利用します。

### 3-1-4 飲食店検索

サイト名	ぐるなび	ホットペッパーグルメ	食べログ
比較内容			
掲載数	約 76,000 店舗約	570,000 店舗	655,905 店舗
利用可能 API	3 種類	3 種類	3 種類

飲食店検索では、「ホットペッパーグルメ」は検索可能な地域が限定されているため記載店舗数が少なく旅行には適していないと判断する。

「ぐるなび」と「食べログ」では「食べログ」の記載店舗数が「ぐるなび」に比べ圧倒的に多いため「食べログ」が適していると判断し利用することにする。

### 3-2 解決に対するアプローチ

まず、デバイスは利用者が増加しているスマートフォンを使うことにします。

スマートフォンを選ぶ理由としては、持ち運びが容易であることと、ユーザーインターフェイスが優れているため利用者が感覚的に使用できることです。

旅行時の無駄を削減するためのものにも関わらず、利用者が使いづらい物では意味がないので利用者が感覚的に利用できるスマートフォンを利用します。

今回はその中から Apple 社の iPhone を利用します。

iPhone を選ぶ理由としては、Android ではそれぞれの機種により細かな調整をこなすことはできませんが、iPhone ではその必要がないためです。

デバイスは iPhone を利用し、アプリケーションを作るための API は「使用サイトの検討」の中で選んだ物を使用します。

## 4. 構成機能・構成要素

### 4-1 構成機能

3章で検討を行った情報行動に基づいて、4章では設計を行っていきます。

- ・ 宿泊予約

楽天トラベルの API を利用し、宿泊施設の検索と予約を行っていきます。

- ・ ルート検索

google map の API を利用することで、徒歩、車、電車でのルート検索を行えるようにします。

- ・ 天気予報

google weather を利用することで、4日間分の天気、気温、湿度、風の強さの情報を手に入れることができます。

- ・ 飲食店検索

食べログの API を利用することで、飲食店の情報を得ることで旅行先での飲食店を探す手助けをします。

### 4-2 構成要素

- ・ 利用者

スマートフォンを利用して、必要に応じて各種情報行動を選択し各サービス提供サイトから情報を取得します。

- ・ サービス提供者

サービス提供者は、宿泊予約、ルート検索、天気予報、飲食店検索の各種サービスの提供を行っているサイトで必要に応じて各サイトから情報を取得していきます。

- ・ 本環境構成要素

本環境は、宿泊利用予約、ルート検索、天気予報、飲食店検索の4つの構成から出来ています。

トップページに4つの情報行動を表示し、旅行時の情報行動を快適に利用できるようにします。

- ・ 設計目標

宿泊予約後に宿泊施設の住所情報を取得し、各種情報に利用できるものにする  
こと  
ルート検索は目的地の情報として利用することで行きルート検索や現地での行動  
後に宿泊施設に戻るまでのルート検索ができます。

天気予報と飲食店検索では、旅行前の準備の段階で利用することができます。

## 5 設計

### 5-1 実装環境

PC 本体	Mac book
CPU	2.4GHz Intel Core 2 Duo
OS	Mac OS X 10.6.8 Snow Leopard
メモリ	2GB 667MHz DDR2 SDRAM
プログラミング言語	HTML, javascript, Objective-C

iPhone 用のアプリケーションを作成するために、Xcode を使用します。

### 5-2 データの取得

データの取得には 4 章で決めた、各サイトから API を利用し取得します。

API とはソフトウェアを開発する際に使用できる命令や関数の集合のことです。

#### 5-2-1 ウェブアプリからネイティブアプリへの変換

Xcode には HTML を組み込み、ネイティブアプリで表示する方法があります。

Resources ファイルの下に使用する HTML で作成したファイルを置きます。

表 5-2-1(1): AccommodationAppDelegate.h

```
#import <UIKit/UIKit.h>

@class WebViewViewController;

@interface AccommodationAppDelegate : NSObject <UIApplicationDelegate> {
    UIWindow *window;
    UIWebView *view;
    AccommodationViewController *viewController;
}

@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) IBOutlet AccommodationViewController
*viewController;

@end
```



NSString \*filepath内のpathForResourceで組み込んだウェブアプリ名を入力し、ofTypeで組み込んだファイルの拡張子を指定しています(表5-2-1(2))。この部分には自動生成されたソースも含まれています。

表5-2-1(2):AccommodationAppDelegate.m

```
#import "AccommodationAppDelegate.h"
#import "AccommodationViewController.h"
@implementation AccommodationAppDelegate
@synthesize window;
@synthesize viewController;
#pragma mark -
#pragma mark Application lifecycle
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    // Set the view controller as the window's root view controller and
display.
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    view = [[UIWebView alloc] initWithFrame:[UIScreen
 mainScreen]applicationFrame]];
    [window addSubview:view];
    NSString *filepath = [[NSBundle mainBundle] pathForResource:@"
reservation(HTMLファイルの名前)"ofType:@"html(ファイルの拡張子)"];
    NSURL *fileurl = [NSURL fileURLWithPath:filepath];
    [view loadRequest:[NSURLRequest
 requestWithURL:fileurl]];
    return YES;
}
- (void)applicationWillResignActive:(UIApplication *)application {
    /*
```

Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.

```
*/  
}
```

```
- (void)applicationDidEnterBackground:(UIApplication *)application {  
    /*
```

Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

If your application supports background execution, called instead of `applicationWillTerminate:` when the user quits.

```
*/  
}
```

```
- (void)applicationWillEnterForeground:(UIApplication *)application {  
    /*
```

Called as part of transition from the background to the inactive state: here you can undo many of the changes made on entering the background.

```
*/  
}
```

```
- (void)applicationDidBecomeActive:(UIApplication *)application {  
    /*
```

Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

```
*/
```

```

}
- (void)applicationWillTerminate:(UIApplication *)application {
    /*
     Called when the application is about to terminate.
     See also applicationWillDidEnterBackground:.
     */
}
#pragma mark -
#pragma mark Memory management
- (void)applicationDidReceiveMemoryWarning:(UIApplication *)application {
    /*
     Free up as much memory as possible by purging cached data objects that
     can be recreated (or reloaded from disk) later.
     */
}
- (void)dealloc {
    [viewController release];
    [view release];
    [window release];
    [super dealloc];
}
@end

```

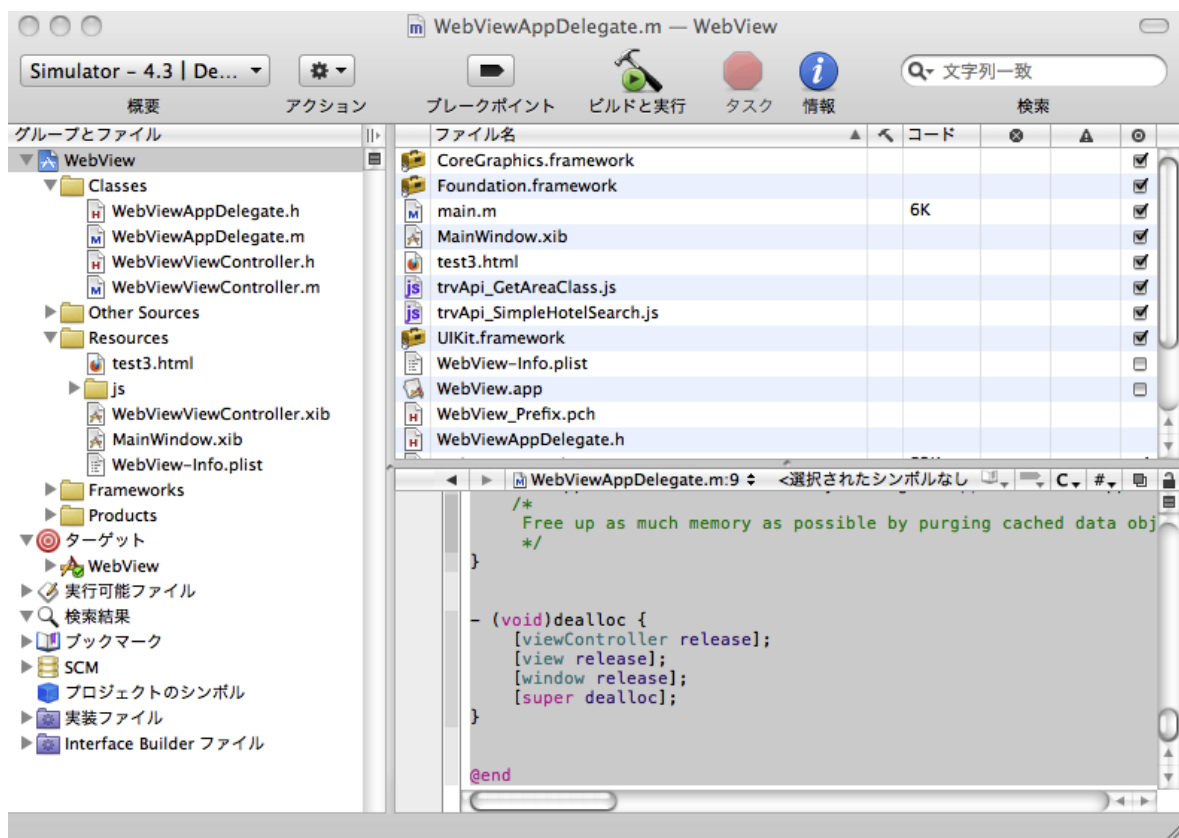


図 5-2-1 Xcode 開発画面

## 5-2-2 宿泊施設予約機能の設計

楽天トラベルの API を利用し、宿泊施設の検索、予約ができることを目的とした、ウェブアプリを作ります。

楽天トラベルで公開されているサンプルプログラム(出典【15】)を参考に作成しました。プログラム内の JavaScript ライブラリ

(callSimpleHotelSearch(), appendParam(query, key), appendCheckParamCsv(query, key), getStar(score), showResult(d), setSmallArea(), setDetailArea(),) は公開されているものをそのまま利用しています。

地区情報はサンプルとして、公開されている trvApi\_GetAreaClass.js を参照し情報を取得しています。

表 5-2-2 reservation.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta http-equiv="Content-Script-Type" content="text/javascript">
<title>宿泊施設予約</title>
</head>
<body>
<script type="text/javascript">
function callSimpleHotelSearch() {
    var developerId = '9e526ecb9ae86aa3aa826653b01e4f3c';
    var affiliateId = '0f0b48de.1efe853a.0f0b48df.7e8c8bd7';
    var callBack = 'showResult';
    var operation = 'SimpleHotelSearch';
    var version = '2009-10-20';
    var requestUrl = 'http://api.rakuten.co.jp/rws/3.0/json?';
    var param = [];
    param.push('developerId='+developerId);
    param.push('callBack='+callBack);
    param.push('operation='+operation);
    param.push('version='+version);
    if(affiliateId) {
        param.push('affiliateId='+affiliateId);
    }
    var query = "";
    query = appendParam(query, "largeClassCode");
    query = appendParam(query, "middleClassCode");
    query = appendParam(query, "smallClassCode");
    query = appendParam(query, "detailClassCode");
    query = appendParam(query, "hotelNo");
    query = appendParam(query, "latitude");
```

```

    query = appendParam(query, "longitude");
    query = appendParam(query, "searchRadius");
    query = appendCheckParamCsv(query, "squeezeCondition");
    query = appendParam(query, "carrier");
    query = appendParam(query, "hits");
    query = appendParam(query, "datumType");
    query = appendParam(query, "hotelThumbnailSize");
    query = appendParam(query, "responseType");
    var scriptTag = document.createElement('script');
    scriptTag.src = requestUrl + param.join('&') + query;
    document.body.appendChild(scriptTag);
}
function appendParam(query, key) {
    var param = document.getElementById(key);
    if(!param || !param.value) {
        return query;
    }
    query += "&" + key + "=" + param.value ;
    return query;
}
function appendCheckParamCsv(query, key) {
    var paramList = document.getElementsByName(key);
    if(paramList == null || paramList.length == 0) {
        return query;
    }
    var param = [];
    for (var i = 0; i < paramList.length; i++) {
        if (paramList[i].checked == true) {
            param.push(paramList[i].value);
        }
    }
}

```

```

    }
    if (0 < param.length) {
        query += "&" + key + "=" + param.join(',');
    }
    return query;
}
function getStar(score) {
    if(!score) {
        return "";
    }
    var fs = parseFloat(score);
    var is = parseInt(score, 10);
    var s = "";
    for(var i = 0; i < is; i++) {
        s += "★";
    }
    if(fs - is > 0.51) {
        s += "★";
    } else if(fs - is > 0.01) {
        s += "☆";
    }
    return s;
}
function showResult(d) {
    var cts = '';
    if(d.Header.Status == "Success") {
        var hotelList = d.Body.SimpleHotelSearch.hotel;
        cts += '<div class="hotelList">';
        for(var i = 0; i < hotelList.length; i++) {
            cts += '<table>';

```

```

        var hotelBasicInfo = hotelList[i].hotelBasicInfo;
        var hotelRatingInfo = hotelList[i].hotelRatingInfo;
        var hotelImageUrl = hotelBasicInfo.hotelThumbnailUrl;
        var hotelInformationUrl = hotelBasicInfo.hotelInformationUrl;
        var hotelName = hotelBasicInfo.hotelName;
        cts += '<tr><td></td>';
        cts += '<td><a href="' + hotelInformationUrl + '">' +
hotelName + '</a>';
            cts += ' <span class="star">' +
getStar(hotelBasicInfo.reviewAverage) + '</span>';
            if(hotelBasicInfo.reviewCount) {
                cts += '(' +hotelBasicInfo.reviewCount+' 件)';
            }
            cts += '<br>' + hotelBasicInfo.access;
            if(hotelBasicInfo.userReview) {
                cts += '<br>【お客様の声】<br>' +
hotelBasicInfo.userReview + "<br>";
            }
            cts += '</td></tr>';
            if(i == 7) {
                break;
            }
            cts += '</table>';
        }
        cts += "</div>";
    } else {
        cts = '<p align="center">' + d.Header.StatusMsg + '</p>';
    }
    document.getElementById('trvResultList').innerHTML = cts;
    document.getElementById('result').style.display = "block";

```



```

}
(function() {
    var developerId = '9e526ecb9ae86aa3aa826653b01e4f3c';
    var affiliateId = '0f0b48de.1efe853a.0f0b48df.7e8c8bd7';
    var callBack = 'createAreaSelect';
    var operation = 'GetAreaClass';
    var version = '2009-03-26';
    var requestUrl = 'http://api.rakuten.co.jp/rws/2.0/json?';
    var param = [];
    param.push('developerId='+developerId);
    param.push('callBack='+callBack);
    param.push('operation='+operation);
    param.push('version='+version);
    if(affiliateId) {
        param.push('affiliateId='+affiliateId);
    }
    var scriptTag = document.createElement('script');
    scriptTag.src = requestUrl + param.join('&');
    document.body.appendChild(scriptTag);
})();
var defaultMiddleArea = 'kanagawa';
var defaultSmallArea = 'hakone';
var rakutenTr_smallAreaData = new Array();
var rakutenTr_datailAreaData = new Array();
function createAreaSelect(data) {
    var prefListTag = '<select id="middleClassCode" name="middleClassCode"
onChange="setSmallArea()"><option value="">指定なし</option>';
    if(data.Header.Status == "Success") {
        var middleClassList =
data.Body.GetAreaClassInfo.largeClass.middleClass;

```

```

        for(var i = 0; i < middleClassList.length; i++) {
            var middleClassCode = middleClassList[i].middleClassCode;
            var selected = "";
            if(middleClassCode == defaultMiddleArea) {
                selected = ' selected="selected"';
            }
            prefListTag += '<option value="' + middleClassCode + '"' +
selected + '>' + middleClassList[i].middleClassName + '</option>';
            rakutenTr_smallAreaData[middleClassCode] =
middleClassList[i].smallClass;
        }
    }
    document.getElementById('middleClassList').innerHTML = prefListTag;
    if(defaultMiddleArea != null && defaultMiddleArea.length != 0) {
        setSmallArea();
    }
}
function setSmallArea() {
    var code = document.getElementById('middleClassCode').value;
    var tag = '<select id="smallClassCode" name="smallClassCode"
onChange="setDetailArea()"><option value="">指定なし</option>';
    if(code != null && code.length != 0) {
        var smallClassList = rakutenTr_smallAreaData[code];
        for(var i = 0; i < smallClassList.length; i++) {
            var selected = "";
            if(document.getElementById('smallClassCode') == null
                && smallClassList[i].smallClassCode ==
defaultSmallArea) {
                selected = ' selected="selected"';
            }
        }
    }
}

```

```

        tag += '<option value="' + smallClassList[i].smallClassCode +
'"" + selected + '>' + smallClassList[i].smallClassName + '</option>';
        if(smallClassList[i].detailClass != null) {
            var detailClassList = smallClassList[i].detailClass;
rakutenTr_datailAreaData[smallClassList[i].smallClassCode] =
detailClassList;
            } else {
                document.getElementById('detailClassList').innerHTML =
"";
            }
        }
        document.getElementById('smallClassList').innerHTML = tag;
        document.getElementById('largeClassCode').value = "japan";
    } else {
        document.getElementById('smallClassList').innerHTML = "";
    }
}
function setDetailArea() {
    var code = document.getElementById('smallClassCode').value;
    if(code != null && code.length != 0) {
        var detailClassList = rakutenTr_datailAreaData[code];
        if(detailClassList != null) {
            var tag = '<select id="detailClassCode"
name="detailClassCode"><option value="">指定なし</option>';
            for(var j = 0; j < detailClassList.length; j++) {
                tag += '<option value="' +
detailClassList[j].detailClassCode + '>' +
detailClassList[j].detailClassName + '</option>';
            }
        }
    }
}

```

```

        document.getElementById('detailClassList').innerHTML = tag;
    } else {
        document.getElementById('detailClassList').innerHTML = "";
    }
}
}
</script>
<form name="rtSearch" action="" method="post"
onSubmit="callSimpleHotelSearch(this);return false">
    <input type="hidden" name="largeClassCode" value="" id="largeClassCode" />
<table cellspacing="1" cellpadding="3">
    <tr>
        <th>地区</th>
        <td>
            <div id="middleClassList"></div>
            <div id="smallClassList"></div>
            <div id="detailClassList"></div>
        </td>
    </tr>
    <tr>
        <th>絞り込み条件</th>
        <td>
            <input type="checkbox" name="squeezeCondition" value="kinen"
id="sq_kinen" /><label for="sq_kinen">禁煙ルーム</label>
            <input type="checkbox" name="squeezeCondition"
value="internet" id="sq_internet" /><label for="sq_internet" />インターネット
トが出来る部屋</label>
            <input type="checkbox" name="squeezeCondition"
value="daiyoku" id="sq_daiyoku" /><label for="sq_daiyoku">大浴場あり</label>

```

```

        <input type="checkbox" name="squeezeCondition"
value="sq_onsen" id="sq_onsen" /><label for="sq_onsen">温泉</label>
    </td>
</tr>
<tr>
    <th>キャリア</th>
    <td>
        <select name="carrier" id="carrier">
            <option value="0">PC</option>
            <option value="1">Mobile</option>
        </select>
    </td>
</tr>
<tr>
    <th>施設画像サイズ</th>
    <td>
        <select name="hotelThumbnailSize" id="hotelThumbnailSize">
            <option value="1">小</option>
            <option value="2" selected="selected">中</option>
            <option value="3">大</option>
        </select>
    </td>
</tr>
</table>
<p align="center"><input type="submit" value=" 検索 " ></p>
</form>
<div id="result">
<div id="trvResultList"></div>
</div>
</body>

```

</html>

地区内の県を指定することで、県内のエリアが変更される。

地区

絞り込み条件  禁煙ルーム  インターネットが出来る部屋  大浴場あり  温泉

キャリア

施設画像サイズ

図 5-2-2(1) reservation.html の表示結果

地区

絞り込み条件  禁煙ルーム  インターネットが出来る部屋  大浴場あり  温泉

キャリア

施設画像サイズ

[宮ノ下温泉 富士屋ホテル](#) ★★★★★☆(1273件)  
箱根登山鉄道「宮ノ下駅」より 徒歩7分 / 小田原厚木道路「箱根口IC」より  
国道1号線経由20分 (11km)  
【お客さまの声】  
サービスの質が非常に高い為何度か利用しておりますが、今回はラウンジの  
店員の対応があまりよくなくとても残念でした。ラウンジに行きケーキセッ  
トを頼みましたが、コーヒーのお代わりがある事前の説明... 2012-01-10  
21:29:21投稿 [つづきはこちら](#)

[箱根ホテル 富士屋ホテルレイクビューアネックス](#) ★★★★★☆(417件)  
東名厚木 I C から箱根新道を経て芦ノ湖大観 I C。そこからわずか5分 / 小  
田急線・箱根湯本駅～バス、またはタクシーで35分  
【お客さまの声】  
歴中のあるホテルで 立地条件もよく 眺めも最高 たが1 箱根の観光は遊

図 5-2-2(2) 検索結果

## 6 動作の検証と評価

### 6-1 動作環境

PC 本体	Mac book
CPU	2.4GHz Intel Core 2 Duo
OS	Mac OS X 10.6.8 Snow Leopard
メモリ	2GB 667MHz DDR2 SDRAM
テスト環境	Xcode, iOS シミュレータ

### 6-2 動作画面

エミュレータ上で、ウェブアプリが動作をするか。

iPhone 環境で結果が表示されるか。

#### 条件項目

- ・ 地区: 県と楽天トラベルで分けられたエリアを指定することで、自分の行きたいエリアの宿泊施設の情報が見られます。
- ・ 絞り込み条件: 泊まる宿泊施設がないと困るもしくは欲しいという物を条件として設定することができます。
- ・ 施設画像サイズ: 検索後の結果で表示される宿泊施設の画像サイズを指定できます。



図 6-2(1) iOS シミュレータ実行結果

表示される施設数は 8 つです。

画像と施設の説明、口コミ、評価が表示されます。





図 6-2(2) 条件設定後の検索結果

選択した施設で泊まる予定日を入力し、宿泊施設に空きがあるか調べることができます。



図 6-2(3) ホテル選択後の結果画

施設検索までしか、作れておらず検索画面が PC の表示画面と変わっていません。

### 6-3 機能評価

今回設計した機能のうち、実装できたものは宿泊予約をするためのアプリのみで、まだ、検索画面が楽天トラベルのサンプルと大差がなく、宿泊施設を検索し予約するところまでしか出来ていません。

その他の機能でルート検索、天気予報、飲食店検索を実装していないため、相互に利用しあうことで出来る非効率な入力動作の簡略化になっていません。

今回設計したものをすべて実装して初めて、今回の目的である煩雑な作業の簡略化がなりたちます。

#### 6-4 考察

旅行の効率化ということを目的として、検討を行ってきたが現時点では効率化できているのかは実証できていない状態で実装部分が予定機能 4 つに対して、1 つの原型しか出来ていない状態である。

今回調べていてわかったことは、API を提供しているサイトはあるのですが有料であったり、使用可能な機能が制限されていて使い勝手良くないということです。まだまだ旅行支援システムとしては体を成していないので、まずは旅行支援システムとして成り立つものにしていきたいです。今後は実装を最優先にしたいのですが実装予定の 4 つの機能では旅行中に調べるためのものになってしまっているので、調べる以外に共有するという機能をつけていきたいと考えています。

今後の課題としては、実装予定である 4 つの機能が相互に利用しあうための、機能構成をどうするのか。どのような情報を利用するのか、そして旅行の支援だけではなく、友人との情報共有をすることができる機能を含めて検討したいと考えています。

## 7 まとめと今後の課題

### 7-1 まとめ

今回旅行に関する情報行動を簡略化することによって、旅行をより有意義にすることを目的として、必要になる情報行動の検討を行い最低限、宿泊施設予約、ルート検索、天気予報、飲食店検索が必要だとわかりました。

必要だと判断した情報行動を利用するために、情報を提供しているサイトから情報量や API の有無を検討しました。

旅行先での利用とユーザーインターフェイスの使いやすさを考え、iPhone を利用したアプリの作成を検討しました。検討した結果情報を提供しているサイトの API を利用して、ウェブアプリとして作成したものをネイティブアプリに変換する方法をとりました。

宿泊予約用のウェブアプリを作成し、ネイティブアプリに変換しました。

予定していた 4 つの機能では検索するのみで、利用頻度としてはあまり高くないと判断し他に SNS のような友人と情報を共有するものが必要だと考えました。

### 7-2 今後の課題

宿泊施設予約を iPhone らしい使いやすいユーザーインターフェイスに整えていきます。

実装予定の 4 つの機能を作成し、4 つの機能で相互に情報を利用できるモデルを考えていきます。

実装予定だった、4 つの機能に加え旅行中に iPhone を使用して撮影した写真を撮影場所の情報を使用して Google プレイスなどを利用してどこで撮った写真なのか、どんな情報があるかなどを表示できればと考えています。

また、情報を Twitter や Facebook などの SNS との連携も考えています。

旅行の準備の段階から旅行終了後まで利用してもらうには、実装予定機能を相互で利用できる仕組みが必要であり、本来の目的にもなっています。

1 つ 1 つの機能を搭載しているだけではなく、相互に利用しあうことによって効率化つながり利用者の無駄が軽減できると考えています。

## 参考文献

- 【1】 じゃらん net  
<http://www.jalan.net/smart/>
- 【2】 楽天トラベル  
<http://travel.rakuten.co.jp/smart/>
- 【3】 JTB  
<http://www.jtb.co.jp/smartphone/>
- 【4】 ジョルダン  
<http://mb.jorudan.co.jp/os/norikae.cgi>
- 【5】 駅探  
<http://sp.ekitan.com/>
- 【6】 NAVITIME  
<http://touch.navitime.co.jp/>
- 【7】 Goo 地図  
<http://map.goo.ne.jp/>
- 【8】 Googlemap  
<http://maps.google.co.jp/>
- 【9】 気象庁  
<http://www.jma.go.jp/jma/index.html>
- 【10】 Yahoo!天気情報  
<http://weather.mobile.yahoo.co.jp/p/weather/top>
- 【11】 weathernews  
<http://weathernews.jp/s/>
- 【12】 ぐるなび  
<http://mobile.gnavi.co.jp/iphone/top/>
- 【13】 ホットペッパーグルメ  
<http://www.hotpepper.jp/smartphone/>
- 【14】 食べログ  
<http://s.tabelog.com/smartphone/top/>
- 【15】 楽天トラベルトップ

<http://travel.rakuten.co.jp/webservice/>

【16】 光行淳子「観光キーワード事典-観光文化への道標-」 学陽書房出版 2009年

## 謝辞

本研究を行うにあたり、お力添え頂いた指導教員の渡辺恭人順教授には感謝いたします。卒業論文提出間際になり、方向転換したにも関わらず懇切丁寧にご指導いただきなんとか卒業論文の提出にこぎつけることができたことを心より感謝申し上げます。拙い文章を書き大変ご迷惑をおかけした事と思います。渡辺恭人順教授の指導により、本来の力以上のものが出来上がりました。最後に、私の卒業論文に関わって下さった方全員にもう一度感謝を述べさせて頂き謝辞とさせていただきます。ありがとうございました。