

4hands

たかくさき療術院様

予約支援システム開発プロジェクト

最終ドキュメント

付録

PM :

江木典之 (日本 IBM)

学生 :

向吉学 (環境情報学部 3 年)

安藤亮一 (環境情報学部 3 年)

真野恵理子 (環境情報学部 2 年)

目次

1. ソースコード	3
1.1. UC01：予約状況を閲覧する（Webアプリケーション）	3
1.1.1. 試作版	3
1.1.2. 製品版	23
1.2. UC02：予約状況を管理する（携帯電話用アプリケーション）	46
1.2.1. 試作版	46
1.2.2. 製品版	74
1.3. UC05 サイト構築	102
2. 進捗報告資料	125

1. ソースコード

1.1. UC01 : 予約状況を閲覧する (Web アプリケーション)

1.1.1. 試作版

[MontlyScheduleManager.class.php]

```
<?php
require_once("../sys_config.php");
require_once("japaneseDate.php");

class MonthlyScheduleManager{
    var $year;
    var $month;
    var $file;
    var $schedule;
    var $holidays;
    var $calendar;

    function MonthlyScheduleManager($year,$month){
        $this->year = $year;
        $this->month = $month;
        $this->calendar = new japaneseDate();
        $this->holidays = $this->calendar->getHolidayList(mktime(0,0,0,$month,1,$year));
        if($month < 10){
            $month = "0".$month;
        }
        $this->schedule = array();
        $spath = "../data/".$year.$month.".dat";
        if(file_exists($spath) && $this->file = fopen($spath,"r")){
            $this->parse();
        }
    }

    /**
     * ファイルを解析する
     */
    function parse(){
        while(!feof($this->file)){
            $scols = fgetcsv($this->file,1000,"");
            $this->putProfiles($scols[4],$scols[1],intval($scols[2]),intval($scols[3]));
        }
    }

    /**
     * 2次元配列に予約か予定を配置する
     *
     * @param unknown_type $profile
     * @param unknown_type $start
     * @param unknown_type $finish
     */
    function putProfiles($profile,$date,$start,$finish){
        if($start >= $finish){
            return;
        }
    }
}
```

```

}else{
    // 現在の開始時間の予約か予定を配列に配置する
    $term = $this->getTerm($start);
    if(!array_key_exists($date,$this->schedule)){
        $this->schedule[$date] = array();
    }
    $this->schedule[$date][$term] = $profile;

    // 開始時間を 30 分増やす
    if($start % 100 == 0){
        $start += 30;
    }else{
        $start += 70;
    }

    // 再帰呼び出し
    $this->putProfiles($profile,$date,$start,$finish);
}
}

/**
 * 時間帯を取得する
 *
 * @param unknown_type $time
 * @return unknown $term
 */
function getTerm($time){
    $term = intval(($time - 800) / 100) * 2 - 1;
    if($time % 100 != 0){
        $term++;
    }
    return $term;
}

/**
 * 指定された時間帯に予約か予定が入っているか調べる
 *
 * @param unknown_type $date
 * @param unknown_type $term
 * @return unknown
 */
function getProfile($date,$term){
    if(array_key_exists($date,$this->schedule)
    array_key_exists($term,$this->schedule[$date]) && $term > 0 && $term <= MAX_TERM){
        return $this->schedule[$date][$term];
    }else{
        return false;
    }
}

/**
 * 指定された日が療術を行っていない日かどうかを調べる
 *
 * @param unknown_type $date

```

```

* @return unknown
*/
function isClosed($date){
    for($i = 1;$i <= MAX_TERM;$i++){
        if($this->getProfile($date,$i) != "plan"){
            return false;
        }
    }
    return true;
}

/**
 * 年を取得する
 *
 * @return unknown
 */
function getYear(){
    return $this->year;
}

/**
 * 月を取得する
 *
 * @return unknown
 */
function getMonth(){
    return $this->month;
}

/**
 * 指定された日が祝日かどうか調べる
 *
 * @param unknown_type $date
 * @return unknown
 */
function isHoliday($date){
    if(array_key_exists($date,$this->holidays)){
        return true;
    }else{
        return false;
    }
}

/**
 * 指定された時間帯が予約の頭かどうかを調べる
 *
 * @param unknown_type $date
 * @param unknown_type $term
 * @return unknown
 */
function isReserveStart($date,$term){
    return $this->getProfile($date,$term) == "reserve" &&
    $this->getProfile($date,$term-1) != "reserve";
}

```

```
/**
 * 指定された時間帯が予約の最後かどうかを調べる
 *
 * @param unknown_type $date
 * @param unknown_type $term
 * @return unknown
 */
function isReserveFinish($date,$term){
    return ($this->getProfile($date,$term) != "reserve" && $this->getProfile($date,$term-1)
== "reserve");
}
}
?>
```

[MonthlyCalendarFactory.class.php]

```
<?php
require_once("../sys_config.php");
require_once("Calendar/Month/Weekdays.php");
require_once("MonthlyScheduleManager.class.php");

class MonthlyCalendarFactory{
    var $year = 0;
    var $month = 0;
    var $date = 0;
    var $schedule;

    function MonthlyCalendarFactory($year,$month,$date = 0){
        $this->year = $year;
        $this->month = $month;
        $this->date = $date;
        $this->schedule = new MonthlyScheduleManager($year,$month);
    }

    function draw(){
        $table = MONTH_TABLE;
        $month_td = MONTH_TD;
        $hol_td = HOLIDAY_TD;
        $sat_td = SATURDAY_TD;
        $def_td = DEFAULT_TD;
        $emp_td = EMPTY_TD;
        $closed_date = CLOSED_DATE;
        $selected_date = SELECTED_DATE;
        $default_date = DEFAULT_DATE;

        $Month = new Calendar_Month_Weekdays($this->year, $this->month, 0);

        $Month->build();

        // テーブルを作る
        $thismonth = $this->month;
        echo "<table class=$table>¥n";
        echo "<tr><td colspan=¥7¥" class=$month_td > $thismonth";
        echo "月 </td></tr>";

        //曜日の列
        echo "<tr><td class=$hol_td>日</td>";
        echo "<td class=$def_td>月</td>";
        echo "<td class=$def_td>火</td>";
        echo "<td class=$def_td>水</td>";
        echo "<td class=$def_td>木</td>";
        echo "<td class=$def_td>金</td>";
        echo "<td class=$sat_td>土</td></tr>";

        while ($Day = $Month->fetch()) {
            if ($Day->isFirst()) {
                echo "<tr>";
            }
        }
    }
}
```

```

if ($Day->isEmpty()) {
    // 空欄のとき
    echo "<td class=$semp_td>&nbsp;</td>¥n";
} else {
    if ($Day->isFirst() || $this->schedule->isHoliday($Day->thisDay())) {
        // 休日のとき
        echo "<td class=$hol_td>";
    } else if ($Day->isLast()) {
        // 土曜のとき
        echo "<td class=$sat_td>";
    } else {
        // 平日のとき
        echo "<td class=$def_td>";
    }

    if ($this->schedule->isClosed($Day->thisDay())) {
        // 療術を行っていないとき
        echo "<p class=$closed_date>".$Day->thisDay()."</p></td>¥n";
    } else if ($Day->thisDay() == $this->date) {
        // 選択された日のとき
        echo
            "
            href=¥"week.php?year=".$Day->thisYear()."&month=".$Day->thisMonth().
            "&date=".$Day->thisDay()."¥"
            class=$selected_date>".$Day->thisDay()."</a></td>¥n";
            "
            target=¥"week¥"
    } else {
        // 通常の場合
        echo
            "
            href=¥"week.php?year=".$Day->thisYear()."&month=".$Day->thisMonth().
            "&date=".$Day->thisDay()."¥"
            class=$default_date>".$Day->thisDay()."</a></td>¥n";
            "
            target=¥"week¥"
    }

    if ($Day->isLast()) {
        echo "</tr>¥n";
    }
}
echo "</table>¥n";
}
?>

```


[WeeklyCalendarFactory.class.php]

```
<?php
require_once("../sys_config.php");
require_once("Calendar/Week.php");
require_once("MonthlyScheduleManager.class.php");

class WeeklyCalendarFactory{
    var $year;
    var $month;
    var $date;
    var $schedule;
    var $next_schedule;
    var $day_of_week = array("日","月","火","水","木","金","土");

    function WeeklyCalendarFactory($year, $month, $date){
        $this->year = $year;
        $this->month = $month;
        $this->date = $date;
    }

    function draw(){
        $table = WEEK_TABLE;
        $head_ym = YEAR_MONTH_HEADER;
        $head_time = TIME_HEADER;
        $hol_row = HOLIDAY_ROW;
        $sat_row = SATURDAY_ROW;
        $def_row = DEFAULT_ROW;
        $date_col = DATE_COL;

        $week = new Calendar_Week($this->year, $this->month, $this->date, 0);
        $week->build();

        print "<table border=¥"0¥" class=$table>¥n";
        print "                " <tr                                class=$head_time><td
style=¥"border-left-width:1px;border-right-width:1px;¥"><img
src=¥"images/time_space.gif¥"></td>¥n";
        for($si = 0;$si < MAX_TERM + 1;$si++){
            // 時間を表示
            if($si % 2 == 0){
                $stime = 9 + ($si / 2);
                if($si < MAX_TERM){
                    print "<td colspan=¥"2¥">".$stime.":00</td>¥n";
                }else{
                    print "                " <td                                colspan=¥"2¥"
style=¥"border-right-width:1px;¥">".$stime.":00</td>¥n";
                }
            }
        }
        print "</tr>¥n";

        // 週別予定表を表示する
        for($si = 0;$day = $week->fetch();$si++){
```

```

    if($this->schedule == null){
        // 最初に月ごとの予定を取得する
        $this->schedule = new
MonthlyScheduleManager($day->thisYear(),$day->thisMonth());
    }else if($this->schedule->getMonth() != $day->thisMonth()){
        // 月をまたいだときに次の月の予定を取得する
        $this->schedule = new
MonthlyScheduleManager($day->thisYear(),$day->thisMonth());
    }

    if($i == 0 || $this->schedule->isHoliday($day->thisDay())){
        // 休日のとき
        print "<tr class=$hol_row>¥n";
    }else if($i == 6){
        // 土曜のとき
        print "<tr class=$sat_row>¥n";
    }else{
        // 平日のとき
        print "<tr class=$def_row>¥n";
    }

    // 日付を表示
    print "                                "<td
class=$date_col>".$day->thisMonth().".$day->thisDay().("$this->day_of_week[$i].")</td>
¥n";
    print "                                style=¥"border-right-width:1px;¥"><img
src=¥"images/no_ap_line.gif¥"></td>¥n";

    for($sterm = 1; $sterm <= MAX_TERM + 1; $sterm++){
        if($sterm % 2 == 0 || $sterm == MAX_TERM + 1){
            // 時間ごとの区切り線
            print "<td style=¥"border-right-width:1px¥">";
        }else{
            print "<td>";
        }

        if($this->schedule->getProfile($day->thisDay(),$sterm) == "reserve"){
            // 予約が入っているとき
            print "<img src=¥"images/ap_line.gif¥">";
        }else if($this->schedule->getProfile($day->thisDay(),$sterm) == "plan"){
            // 療術を行っていないとき
            print "<img src=¥"images/no_op_line.gif¥">";
        }else{
            // 空いているとき
            print "<img src=¥"images/no_ap_line.gif¥">";
        }
    }

    print "</td></tr>¥n";
}

print "</table>¥n";
}
}

```

[DaylyCalendarFactory.class.php]

```
<?php
```

```
require_once("../sys_config.php");  
require_once("Calendar/Month/Weekdays.php");  
require_once("MonthlyScheduleManager.class.php");
```

```
class DaylyCalendarFactory{
```

```
    var $year;  
    var $month;  
    var $date;  
    var $schedule;
```

```
function DaylyCalendarFactory($year,$month,$date){  
    settype($year,"string");  
    settype($month,"string");  
    settype($date,"string");  
    $this->year = $year;  
    $this->month = $month;  
    $this->date = $date;  
    $this->schedule = new MonthlyScheduleManager($year,$month);  
}
```

```
function draw(){  
    print "                <br>¥n";
```

```
    for($term = 1; $term <= MAX_TERM + 1; $term++){  
        $time = 9 + intval(($term / 2));  
        $timeline = false;
```

```
        if($term % 2 == 1){  
            if($time < 10){  
                print "                ";  
            }else{  
                print " | ";  
            }  
        }
```

```
        print $time.":00  ";
```

```
        $timeline = true;  
    }else{  
        print "                | ";  
    }  
}
```

```
// 予約か予定が入っているときマークをつける
```

```
if($this->schedule->getProfile($this->date,$term) == "reserve" ||  
$this->schedule->isReserveFinish($this->date,$term)){  
    if($this->schedule->isReserveStart($this->date,$term)){  
        print "                ";  
    }else if($this->schedule->isReserveFinish($this->date,$term)){  
        print "                ";  
    }  
}
```

```

    }else{
        print "      ";
    }
}else if($this->schedule->getProfile($this->date,$term) == "plan"){
    print " / / / ";
}else{
    print "      ";
}

if($timeline){
    print " ";
}else{
    print " | ";
}

print "<br>¥n";
}

print "          <br>¥n";
}
}

```

[sys_config.php]

```
<?php
set_include_path("../includes" . PATH_SEPARATOR . "../includes/PEAR" .
PATH_SEPARATOR . "." . PATH_SEPARATOR . get_include_path());
mb_internal_encoding("UTF-8");
mb_http_output("Shift_JIS");

define("MAX_TERM",24);

define("MONTH_TABLE","¥calendar¥");
define("MONTH_TD","¥month¥");
define("HOLIDAY_TD","¥holiday¥");
define("SATURDAY_TD","¥saturday¥");
define("DEFAULT_TD","¥default_td¥");
define("EMPTY_TD","¥empty_td¥");
define("CLOSED_DATE","¥closed¥");
define("SELECTED_DATE","¥selected¥");
define("DEFAULT_DATE","¥default_date¥");

define("WEEK_TABLE","¥week_table¥");
define("YEAR_MONTH_HEADER","¥head_ym¥");
define("TIME_HEADER","¥head_time¥");
define("HOLIDAY_ROW","¥hol_row¥");
define("SATURDAY_ROW","¥sat_row¥");
define("DEFAULT_ROW","¥def_row¥");
define("DATE_COL","¥date¥");

?>
```

[update.php]

```
<?php

define("SUCCESS",1);
define("ERROR",0);

// POST の内容を GET にコピーする
if(isset($_POST["mode"])){
    $_GET = $_POST;
}

if(isset($_GET["mode"])){
    // データが送られた場合
    switch($_GET["mode"]){
        case "insert":
            // データ挿入の場合
            print insertData($_GET);
            break;
        case "edit":
            // データ編集の場合
            print editData($_GET);
            break;
        case "delete":
            // データ削除の場合
            print deleteData($_GET);
            break;
        default:
            // それ以外の場合
            print ERROR;
    }
}
else{
    // データが送られなかった場合
    print ERROR;
}

/**
 * 新規レコードを挿入する
 */
function insertData($data,$setid = 0){
    if(!validate($data)){
        // パラメータの妥当性を調べる
        return ERROR;
    }
    // レコードの ID を設定する
    if($setid == 0){
        $sid = getCurrentID() + 1;
    }
    else{
        $sid = $setid;
    }

    // ファイルを開く（追記）
    $file = fopen("./data/".$data["year"].$data["month"].".dat","a");
    $stable_file = fopen("./data/id_table.dat","a");
```

```

// レコードを作製する (ID,日付,開始時間,終了時間,予約か予定)
$record = $id.", ".$data["date"].", ".$data["start"].", ".$data["finish"].", ".$data["profile"]."¥n";
$stable_col = $id.", ".$data["year"].$data["month"]."¥n";

// レコードを書き込む
if(fwrite($file,$record) && fwrite($stable_file,$stable_col)){
    fclose($file);
    fclose($stable_file);
}else{
    return ERROR;
}

return $id;
}

/**
 * レコードを編集する
 */
function editData($data){

    // 古いレコードを削除する
    if(!deleteData($data)){
        return ERROR;
    }

    // 新しいレコードを挿入する
    if(!insertData($data,$data["id"])){
        return ERROR;
    }

    return SUCCESS;
}

/**
 * レコードを削除する
 */
function deleteData($data){
    if(!validate($data)){
        return ERROR;
    }

    $stable = getIDTable();

    // 既に ID が設定されているかどうか調べる
    if(!array_key_exists(intval($data["id"]),$stable)){
        return ERROR;
    }

    // 予約情報ファイルからレコードを削除する
    $file = fopen("./data/".$stable[$data["id]].".dat", "r");
    $file_lines = array();
    while(!feof($file)){
        $cols = fgetcsv($file,100000,",");

```

```

        if($cols[0] != $data["id"] && $cols[0] != ""){
            array_push($file_lines,$cols);
        }
    }
    fclose($file);

    $file = fopen("./data/".$table[$data["id]].".dat","w+");
    foreach($file_lines as $line){
        fwrite($file,implode(",",$line)."¥n");
    }
    fclose($file);

    // ID 表からレコードを削除する
    $file = fopen("./data/id_table.dat","w+");
    foreach($table as $key => $value){
        if($key != $data["id"] && $key != ""){
            fwrite($file,$key.", ".$value."¥n");
        }
    }

    return true;
}

/**
 * 値の妥当性を調べる
 *
 * @param unknown_type $data
 * @return unknown
 */
function validate(&$data){
    if($data["mode"] != "insert"){
        if(!isset($data["id"])){
            return false;
        }
    }
    if($data["mode"] != "delete"){
        if(!isset($data["year"]) || !isset($data["month"]) || !isset($data["date"])
        || !isset($data["start"]) || !isset($data["finish"]) || !isset($data["profile"])){
            return false;
        }else{
            if(!checkdate(intval($data["month"]),intval($data["date"]),intval($data["year"]))){
                return false;
            }
            if(intval($data["start"]) < 900 || intval($data["start"]) > 2100 ||
            intval($data["finish"]) < 900 || intval($data["finish"]) > 2100){
                return false;
            }
            if($data["profile"] != "plan" && $data["profile"] != "reserve"){
                return false;
            }
        }
    }
    if($data["month"] < 10){
        $data["month"] = "0".$data["month"];
    }
}

```



```

    }
}

return true;
}

/**
 * ID と予約情報ファイル(年月)の対応表を取得する
 */
function getIDTable(){
    $id_table = array();
    if(file_exists("./data/id_table.dat")){
        $file = fopen("./data/id_table.dat","r");
        while(!feof($file)){
            $line = fgetcsv($file,100000,",");
            if($line[0] != ""){
                $id_table[$line[0]] = $line[1];
            }
        }
        fclose($file);
    }
    return $id_table;
}

/**
 * 現在使われている ID の数を取得する
 */
function getCurrentID(){
    $id_table = getIDTable();
    $current_key = 0;
    foreach($id_table as $key => $value){
        $key = intval($key);
        if($key > $current_key){
            $current_key = $key;
        }
    }
    return $current_key;
}
?>

```

[calendar.php]

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<style type="text/css">
<!--
.left{
    text-align:center;
}
.left table{
    margin:10px auto;
    border-collapse:collapse;
}

.left table tr td{
    text-align:center;
    vertical-align:middle;
    border:1px solid #aaaaaa;
    font-size:0.8em;
    margin:0px;
    padding:3px;
}

a:link{ color:#0000FF;text-decoration:none; }
a:hover{ color:#FFFF00;text-decoration:none; }
a:active{ color:#0000FF;text-decoration:none; }
a:visited{ color:#0000FF;text-decoration:none; }

.holiday{
    background-color:#ffcccc;
}

.saturday{
    background-color:#eeeeee;
}

.default_td{
    background-color:#f5fff;
}

.empty_td{
    background-color:#ffffff;
}

.closed{
    color:#cccccc;
}

.selected{
    font-weight:bold;
}
-->
</style>
</head>
```

```
<body>
<div class="container">

<div class="left">

<?php
require_once("../sys_config.php");
require_once("MonthlyCalendarFactory.class.php");

$year = intval(date("Y"));
$month = intval(date("m"));
$date = intval(date("j"));

for($i = 0;$i < 3;$i++){
    if($i == 0){
        $calendar = new MonthlyCalendarFactory($year,$month,$date);
    }else{
        $calendar = new MonthlyCalendarFactory($year,$month);
    }
    $calendar->draw();

    $month += 1;
    if($month > 12){
        $month -= 12;
        $year++;
    }
}

?>

</div>
</body>
</html>
```

[week.php]

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<style = "text/css">
<!--
.right{
    text-align:center;
}

table {

    border-collapse:collapse;
    margin:10px auto;
}

table tr td{
    margin:0px;
    padding:0px;
    text-align:center;
    border-style:solid;
    border-color:#aaaaaa;
    border-top-width:1px;
    border-bottom-width:1px;
    border-left-width:0px;
    border-right-width:0px;
    font-size:0.8em;
}

.date{
    border-left-width:1px;
    border-right-width:1px;
    padding:0px 3px;
}

.head_ym th{
    padding:20px 0px;
    border-style:none;
    font-size:14pt;
    font-weight:bold;
}

.head_time td{
    background-image:URL("images/time_back.gif");
}

.hol_row{
    background-color:#ffcccc;
}

.sat_row{
    background-color:#eeeeee;
}
```

```
.def_row{
  background-color:#f5ffff;
}
-->
</style>
</head>
<body>
<div class="right">
<?php
require_once("../sys_config.php");
require_once("WeeklyCalendarFactory.class.php");

if(isset($_GET["year"]) && isset($_GET["month"]) && isset($_GET["date"])){
  $week = new WeeklyCalendarFactory($_GET["year"],$_GET["month"],$_GET["date"]);
}else{
  $week = new WeeklyCalendarFactory(date('Y'),date('m'),date('j'));
}
$week->draw();

?>
</div>
</body>
</html>
```

[date.php]

```
<?php
require_once("../sys_config.php");
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>療術院予約状況</title>
</head>
<body>
<h1>たかくさき療術院</h1><br>
<h2>予約状況参照</h2><br>
<p>参照したい日付を選択してください。 </p>

<?php
if(isset($_GET["err"])){
    print "<p><font color=¥"#FF0000¥">日付の値が正しくありません。 </font></p>";
}
?>
<form method="GET" action="yoyaku_joukyou.php">
<?php
print "<input type=¥"text¥" maxlength=¥"4¥" size=¥"4¥" name=¥"year¥" value=" .
date('Y') . " istyle=¥"4¥" mode=¥"numeric¥">年<br>¥n";
print "<input type=¥"text¥" maxlength=¥"2¥" size=¥"2¥" name=¥"month¥" value=" .
date('m') . " istyle=¥"4¥" mode=¥"numeric¥">月";
print "<input type=¥"text¥" maxlength=¥"2¥" size=¥"2¥" name=¥"date¥" value=" .
date('d') . " istyle=¥"4¥" mode=¥"numeric¥">日<br>¥n";
?>
<input type="submit" value="参照">
</form>
</body>
</html>
```

1.1.2. 製品版

[MontlyScheduleManager.class.php]

```
<?php
require_once("sys_config.php");
require_once("japaneseDate.php");

/**
 * 毎月の予約状況を管理するクラス
 *
 * description:
 * 月ごとにファイルで管理されている予約状況を解析して、予約状況を参照するためのいく
つかの
 * メソッドを提供する。
 */
class MonthlyScheduleManager{
    var $year;
    var $month;
    var $file;
    var $schedule;
    var $holidays;
    var $calendar;

    function MonthlyScheduleManager($year,$month){
        $this->year = $year;
        $this->month = $month;
        $this->calendar = new japaneseDate();
        $this->holidays = $this->calendar->getHolidayList(mktime(0,0,0,$month,1,$year));
        if($month < 10 && strlen($month) < 2){
            // 月の文字列を変換する ex) 3 03
            $month = "0".$month;
        }
        $this->schedule = array();
        $spath = "../data/".$year.$month.".dat";
        if(file_exists($spath) && $this->file = fopen($spath,"r")){
            $this->parse();
        }
    }

    /**
     * ファイルを解析する
     */
    function parse(){
        while(!feof($this->file)){
            $cols = fgetcsv($this->file,1000,",");
            $this->putProfiles(intval($cols[1]),intval($cols[2]),intval($cols[3]));
        }
    }

    /**
     * 2次元配列に予約を配置する
     *
     * @param unknown_type $date
     */
}
```

```

* @param unknown_type $start
* @param unknown_type $finish
*/
function putProfiles($date,$start,$finish){
    if($start >= $finish){
        return;
    }else{
        // 現在の開始時間の予約を配列に配置する
        $term = $this->getTerm($start);
        if(!array_key_exists($date,$this->schedule)){
            $this->schedule[$date] = array();
        }
        $this->schedule[$date][$term] = 1;

        // 開始時間を 30 分増やす
        if($start % 100 == 0){
            $start += 30;
        }else{
            $start += 70;
        }

        // 再帰呼び出し
        $this->putProfiles($date,$start,$finish);
    }
}

/**
 * 時間帯を取得する
 *
 * @param unknown_type $time
 * @return unknown $term
 *
 * description:
 * 9 時からの 30 分間を時間帯 1、9 時半からの 30 分間を時間帯 2 としたときに、引数で
 * 渡された時間で始まる時間帯がどの時間帯なのかを返す。
 */
function getTerm($time){
    $term = intval(($time - 800) / 100) * 2 - 1;
    if($time % 100 != 0){
        $term++;
    }
    return $term;
}

/**
 * 指定された時間帯に予約か予定が入っているか調べる
 *
 * @param unknown_type $date
 * @param unknown_type $term
 * @return unknown
 */
function getProfile($date,$term){
    if(array_key_exists($date,$this->schedule)
        array_key_exists($term,$this->schedule[$date]) && $term > 0 && $term <= MAX_TERM){

```



```

        return $this->schedule[$date][$term];
    }else{
        return false;
    }
}

/**
 * 指定された日が療術を行っていない日かどうかを調べる
 *
 * @param unknown_type $date
 * @return unknown
 */
function isClosed($date){
    for($i = 1;$i <= MAX_TERM;$i++){
        if($this->getProfile($date,$i) != 1){
            return false;
        }
    }
    return true;
}

/**
 * 年を取得する
 *
 * @return unknown
 */
function getYear(){
    return $this->year;
}

/**
 * 月を取得する
 *
 * @return unknown
 */
function getMonth(){
    return $this->month;
}

/**
 * 指定された日が祝日かどうか調べる
 *
 * @param unknown_type $date
 * @return unknown
 */
function isHoliday($date){
    if(array_key_exists($date,$this->holidays)){
        return true;
    }else{
        return false;
    }
}

/**

```

```

* 指定された時間帯が予約の始まりにあたるかどうかを調べる
*
* @param unknown_type $date
* @param unknown_type $term
* @return unknown
*/
function isReserveStart($date,$term){
    return $this->getProfile($date,$term) == 1 && $this->getProfile($date,$term-1) != 1;
}

/**
* 指定された時間帯が予約の終わりにあたるかどうかを調べる
*
* @param unknown_type $date
* @param unknown_type $term
* @return unknown
*/
function isReserveFinish($date,$term){
    return ($this->getProfile($date,$term) == 1 && $this->getProfile($date,$term+1) != 1);
}
}
?>

```

[MonthlyCalendarFactory.class.php]

```
<?php
require_once("../sys_config.php");
require_once("Calendar/Month/Weekdays.php");
require_once("MonthlyScheduleManager.class.php");

/**
 * カレンダーを作成・表示するクラス
 */
class MonthlyCalendarFactory{
    var $year = 0;
    var $month = 0;
    var $date = 0;
    var $schedule;

    function MonthlyCalendarFactory($year,$month,$date = 0){
        $this->year = $year;
        $this->month = $month;
        $this->date = $date;
        $this->schedule = new MonthlyScheduleManager($year,$month);
    }

    function draw(){
        $table = MONTH_TABLE;
        $month_td = MONTH_TD;
        $today_td = TODAY_TD;
        $hol_td = HOLIDAY_TD;
        $sat_td = SATURDAY_TD;
        $def_td = DEFAULT_TD;
        $emp_td = EMPTY_TD;
        $closed_date = CLOSED_DATE;
        $default_date = DEFAULT_DATE;

        $Month = new Calendar_Month_Weekdays($this->year, $this->month, 0);

        $Month->build();

        // テーブルを作る
        $thismonth = $this->month;
        echo "<table class=$table>¥n";
        echo "<tr><td colspan=¥7¥" class=$month_td > $thismonth";
        echo "月 </td></tr>";

        //曜日の列
        echo "<tr><td class=$hol_td>日</td>";
        echo "<td class=$def_td>月</td>";
        echo "<td class=$def_td>火</td>";
        echo "<td class=$def_td>水</td>";
        echo "<td class=$def_td>木</td>";
        echo "<td class=$def_td>金</td>";
        echo "<td class=$sat_td>土</td></tr>";

        while ($Day = $Month->fetch()) {
            if ($Day->isFirst()) {
```

```

    echo "<tr>";
}

if ($Day->isEmpty()) {
    // 空欄のとき
    echo "<td class=$semp_td>&nbsp;</td>¥n";
} else {
    if($Day->thisDay() == $this->date){
        // 今日の時
        echo "<td class=$today_td>";
    } else if ($Day->isFirst() || $this->schedule->isHoliday($Day->thisDay())) {
        // 休日のとき
        echo "<td class=$hol_td>";
    } else if($Day->isLast()){
        // 土曜の時
        echo "<td class=$sat_td>";
    } else{
        // 平日の時
        echo "<td class=$def_td>";
    }

    if($this->schedule->isClosed($Day->thisDay())){
        // 療術を行っていないとき
        echo "<p class=$closed_date>".$Day->thisDay()."</p></td>¥n";
    } else{
        // 通常の時
        echo
href=¥"yoyaku_joukyou.php?year=".$Day->thisYear()."&month=".$Day->thisMonth().
        "&date=".$Day->thisDay()."¥"
class=$default_date> ".$Day->thisDay()."</a></td>¥n";
    }
}

    if ($Day->isLast()) {
        echo "</tr>¥n";
    }
}
}

echo "</table>¥n";
}
?>

```

[WeeklyCalendarFactory.class.php]

```
<?php
require_once("../sys_config.php");
require_once("Calendar/Week.php");
require_once("MonthlyScheduleManager.class.php");

/**
 * 週別予約状況表を作成・表示するクラス
 * for PC Browser
 */
class WeeklyCalendarFactory{
    var $year;
    var $month;
    var $date;
    var $schedule;
    var $next_schedule;
    var $day_of_week = array("日","月","火","水","木","金","土");

    function WeeklyCalendarFactory($year, $month, $date){
        $this->year = $year;
        $this->month = $month;
        $this->date = $date;
    }

    function draw(){
        $table = WEEK_TABLE;
        $head_ym = YEAR_MONTH_HEADER;
        $head_time = TIME_HEADER;
        $hol_row = HOLIDAY_ROW;
        $sat_row = SATURDAY_ROW;
        $def_row = DEFAULT_ROW;
        $selected_date_col = SELECTED_DATE_COL;
        $date_col = DATE_COL;

        $week = new Calendar_Week($this->year, $this->month, $this->date, 0);
        $week->build();

        print "<table border=¥"0¥" class=$table>¥n";
        print "                <tr                                class=$head_time><td
style=¥"border-left-width:1px;border-right-width:1px;¥"><img
src=¥"images/time_space.png¥"></td>¥n";
        for($si = 0; $si < MAX_TERM + 1; $si++){
            // 時間を表示
            if($si % 2 == 0){
                $stime = 9 + ($si / 2);
                if($si < MAX_TERM){
                    print "<td colspan=¥"2¥">".$stime.":00</td>¥n";
                }else{
                    print "                <td                                colspan=¥"2¥"
style=¥"border-right-width:1px;¥">".$stime.":00</td>¥n";
                }
            }
        }
    }
}
```

```

}
print "</tr>¥n";

// 週別予定表を表示する
for($i = 0;$day = $week->fetch();$i++){
    if($this->schedule == null){
        // 最初に月ごとの予定を取得する
        $this->schedule = new
MonthlyScheduleManager($day->thisYear(),$day->thisMonth());
    }else if($this->schedule->getMonth() != $day->thisMonth()){
        // 月をまたいだときに次の月の予定を取得する
        $this->schedule = new
MonthlyScheduleManager($day->thisYear(),$day->thisMonth());
    }

    if($i == 0 || $this->schedule->isHoliday($day->thisDay())){
        // 休日のとき
        print "<tr class=$hol_row>¥n";
    }else if($i == 6){
        // 土曜のとき
        print "<tr class=$sat_row>¥n";
    }else{
        // 平日のとき
        print "<tr class=$def_row>¥n";
    }

    // 日付を表示
    if($day->thisMonth() == $this->month && $day->thisDay() == $this->date){
        print "                " <td
class=$selected_date_col>".$day->thisMonth().".".$day->thisDay().(".$this->day_of_week[$i
].")</td>¥n";
    }else{
        print "                " <td
class=$date_col>".$day->thisMonth().".".$day->thisDay().(".$this->day_of_week[$i].")</td>
¥n";
    }
    print "                " <td
style=¥"border-right-width:1px;¥"><img
src=¥"images/no_ap_line.png¥"></td>¥n";

    for($sterm = 1; $sterm <= MAX_TERM + 1; $sterm++){
        if($sterm % 2 == 0 || $sterm == MAX_TERM + 1){
            // 時間ごとの区切り線
            print "<td style=¥"border-right-width:1px¥">";
        }else{
            print "<td>";
        }
    }

    if($this->schedule->getProfile($day->thisDay(),$sterm)){
        // 予約や予定が入っているとき
        print "<img src=¥"images/ap_line.png¥">";
    }else{
        // 空いているとき
        print "<img src=¥"images/no_ap_line.png¥">";
    }
}

```

```
    }  
    print "</td></tr>¥n";  
  }  
  print "</table>¥n";  
}  
}
```

[WeeklyCalendarImageFactory.class.php]

```
<?php
require_once("../sys_config.php");
require_once("Calendar/Week.php");
require_once("MonthlyScheduleManager.class.php");

/**
 * 週別予約状況表の画像を作成するクラス
 *
 * description:
 * 携帯電話での週別予約状況参照のために、週ごとの予約状況を表形式にして描画した画像
リソースを出力する。
 * このクラスを呼び出す側は、出力された画像リソースから、フォーマットにしたがって画像
の生成を行う。
 */
class WeeklyCalendarImageFactory{

    var $year;
    var $month;
    var $date;

    function WeeklyCalendarImageFactory($year,$month,$date){
        $this->year = $year;
        $this->month = $month;
        $this->date = $date;
    }

    /**
     * 画像を描画して取得する
     *
     */
    function getImage(){

        // キャンバスを作成
        $image = imagecreate(WIDTH,HEIGHT);

        // 背景色を設定
        $background = imagecolorallocate($image, 255, 255, 255);

        // 枠組みとなる表を描画する
        $this->drawTable($image);

        // 日付の表示を描画する
        $this->drawDate($image);

        // 選択された日付に強調枠を描画する
        $this->drawAccentFrame($image);

        // 予約が入っているところに線を描画する
        $this->drawReserveBar($image);

        return $image;
    }
}
```



```

/**
 * 枠組みとなる表を描画する
 *
 * @param unknown_type $image
 */
function drawTable(&$image){
    // 色情報を作成
    $black = imagecolorallocate($image, 0, 0, 0);

    // 表の上に時間を描画する
    $y = TABLE_TOP_Y - (imagefontheight(3) + 1);
    for($i = 0; $i < 5; $i++){
        $time = 9 + $i * 3;
        $fontwidth = imagefontwidth(3);
        if($time >= 10){
            $fontwidth *= 2;
        }
        $x = TABLE_LEFT_X + (CELL_WIDTH * 3 * $i) - intval($fontwidth / 2);
        imagestring($image, FONT, $x, $y, $time, $black);
    }

    // 表の横線を描画する
    for ($i = 0; $i < CELL_COL + 1; $i++) {
        $base = CELL_HEIGHT * $i;
        imageline($image, TABLE_LEFT_X, TABLE_TOP_Y + $base, TABLE_LEFT_X +
        (CELL_ROW * CELL_WIDTH), TABLE_TOP_Y + $base, $black);
    }

    // 表の縦線を描画する
    for ($i = 0; $i < CELL_ROW + 1; $i++) {
        $base = CELL_WIDTH * $i;
        imageline($image, TABLE_LEFT_X + $base, TABLE_TOP_Y, TABLE_LEFT_X +
        $base, TABLE_TOP_Y + (CELL_COL * CELL_HEIGHT), $black);
    }

    // 表の縦破線を描画する
    for ($i = 0; $i < CELL_ROW; $i++) {
        $base = CELL_WIDTH * $i;
        imagedashedline($image, TABLE_LEFT_X + $base + (CELL_WIDTH / 2),
        TABLE_TOP_Y, TABLE_LEFT_X + $base + (CELL_WIDTH / 2), TABLE_TOP_Y +
        (CELL_COL * CELL_HEIGHT), $black);
    }
}

/**
 * 日付の表示を描画する
 *
 * @param unknown_type $image
 */
function drawDate(&$image){
    // 色情報を作成

```

```

$black = imagecolorallocate($image,0,0,0);
$red = imagecolorallocate($image,255,0,0);
$blue = imagecolorallocate($image,0,0,255);

// その日を含む週別のカレンダーを取得
$week = new Calendar_Week($this->year, $this->month, $this->date, 0);
$week->build();

$schedule = null;
$start_month = null;
$day = null;
for($i = 0;$i < 7;$i++){
    $day = $week->fetch();
    if(!isset($schedule)){
        // 最初に月ごとの予定を取得する
        $schedule = new MonthlyScheduleManager($day->thisYear(),$day->thisMonth());
    }else if($schedule->getMonth() != $day->thisMonth()){
        // 月をまたいだときに次の月の予定を取得する
        $schedule = new MonthlyScheduleManager($day->thisYear(),$day->thisMonth());
    }
    // 週の最初のの月を取得する
    $start_month = $day->thisMonth();

    // 休日・土曜日・平日で色分けをする
    $color;
    if($i == 0 || $schedule->isHoliday($day->thisDay())){
        // 休日の時
        $color = $red;
    }else if($i == 6){
        // 土曜日のとき
        $color = $blue;
    }else{
        // 平日のとき
        $color = $black;
    }

    // 日付を描画する
    $date_x = DATE_LEFT_X;
    $date_y = DATE_TOP_Y + ($i * CELL_HEIGHT + 1);
    imagestring($image,FONT+1,$date_x,$date_y, $day->thisMonth() . "/" .
$day->thisDay(),$color);
}
}

/**
 * 選択された日付に強調枠を表示する
 *
 * @param unknown_type $image
 */
function drawAccentFrame(&$image){

    // 色情報を作成
    $green = imagecolorallocate($image,0, 200, 0);

```

```

// その日を含む週別のカレンダーを取得
$week = new Calendar_Week($this->year, $this->month, $this->date, 0);
$week->build();

for($i = 0;$day = $week->fetch();$i++){
    if($day->thisMonth() == $this->month && $day->thisDay() == $this->date){
        // 強調枠を表示する
        imagerectangle($image, TABLE_LEFT_X, TABLE_TOP_Y + (CELL_HEIGHT * $i),
            TABLE_LEFT_X + (CELL_ROW * CELL_WIDTH), TABLE_TOP_Y + (CELL_HEIGHT * ($i
            + 1)), $green);
        imagerectangle($image, TABLE_LEFT_X - 1, TABLE_TOP_Y + (CELL_HEIGHT *
            $i) - 1, TABLE_LEFT_X + (CELL_ROW * CELL_WIDTH) + 1, TABLE_TOP_Y +
            (CELL_HEIGHT * ($i + 1)) + 1, $green);
        }
    }
}

/**
 * 予約が入っているところに線を描画する
 *
 * @param unknown_type $image
 */
function drawReserveBar(&$image){
    // 色情報を作成
    $blue = imagecolorallocate($image, 0, 0, 255);

    // その日を含む週別のカレンダーを取得
    $week = new Calendar_Week($this->year, $this->month, $this->date, 0);
    $week->build();

    $schedule = null;

    for($i = 0;$day = $week->fetch();$i++){
        if(!isset($schedule)){
            // 最初に月ごとの予約状況を取得する
            $schedule = new MonthlyScheduleManager($day->thisYear(), $day->thisMonth());
        } else if($schedule->getMonth() != $day->thisMonth()){
            // 週の中で月をまたいだときに次の月の予約状況を取得する
            $schedule = new MonthlyScheduleManager($day->thisYear(), $day->thisMonth());
        }
        for($term = 1; $term <= MAX_TERM; $term++){
            if($schedule->getProfile($day->thisDay(), $term)){
                // 予約が入っているとき
                $x = TABLE_LEFT_X + (CELL_WIDTH / 2) * ($term - 1);
                $y = TABLE_TOP_Y + (CELL_HEIGHT - BAR_WIDTH) / 2 + (CELL_HEIGHT *
                $i);
                imagefilledrectangle($image, $x, $y, $x + (CELL_WIDTH / 2), $y +
                BAR_WIDTH, $blue);
            }
        }
    }
}
?>

```

[sys_config.php]

```
<?php
set_include_path("../includes" . PATH_SEPARATOR . "../includes/PEAR" .
PATH_SEPARATOR . "../mobile" . PATH_SEPARATOR . "../pc" . PATH_SEPARATOR . ".." .
PATH_SEPARATOR . get_include_path());
mb_internal_encoding("UTF-8");
mb_http_output("Shift_JIS");
putenv('GDFONTPATH=' . realpath('.'));

$domain = "http://for-hands.crew.sfc.keio.ac.jp/reserve";

define("MAX_TERM",28);
define("START_TIME",900);
define("FINISH_TIME",2300);

// CSS 用クラス名
define("MONTH_TABLE","¥calendar¥");
define("MONTH_TD","¥month¥");
define("TODAY_TD","¥today¥");
define("HOLIDAY_TD","¥holiday¥");
define("SATURDAY_TD","¥saturday¥");
define("DEFAULT_TD","¥default_td¥");
define("EMPTY_TD","¥empty_td¥");
define("CLOSED_DATE","¥closed¥");
define("DEFAULT_DATE","¥default_date¥");

define("WEEK_TABLE","¥week_table¥");
define("YEAR_MONTH_HEADER","¥head_ym¥");
define("TIME_HEADER","¥head_time¥");
define("HOLIDAY_ROW","¥hol_row¥");
define("SATURDAY_ROW","¥sat_row¥");
define("DEFAULT_ROW","¥def_row¥");
define("SELECTED_DATE_COL","¥selected¥");
define("DATE_COL","¥date¥");

define("FONT",2); // フォントのサイズ
define("WIDTH",240); // キャンパスの幅
define("HEIGHT",140); // キャンパスの高さ
define("LEFT_X",1); // 描画を開始する X 座標
define("TOP_Y",5); // 描画を開始する Y 座標
define("TABLE_LEFT_X",LEFT_X + 38); // 表の描画を開始する X 座標
define("TABLE_TOP_Y",TOP_Y + 10); // 表の描画を開始する Y 座標
define("CELL_WIDTH",14); // 表の 1 マスの幅
define("CELL_HEIGHT",imagefontheight(3) + 3); // 表の 1 マスの高さ
define("CELL_ROW",14); // 表の列数
define("CELL_COL",7); // 表の行数
define("DATE_LEFT_X",LEFT_X); // 日付の描画を開始する X 座標
define("DATE_TOP_Y", TABLE_TOP_Y); // 日付の描画を開始する Y 座標
define("BAR_WIDTH",CELL_HEIGHT - 8); // 予約の線の幅

define("SUCCESS",1); // 通信成功時の戻り値
define("ERROR",0); // 通信失敗時の戻り値

?>
```

[update.php]

```
<?php

require_once("sys_config.php");

// POST の内容を GET にコピーする
if(isset($_POST["mode"])){
    $_GET = $_POST;
}

if(isset($_GET["mode"])){
    // データが送られた場合
    switch($_GET["mode"]){
        case "insert":
            // データ挿入の場合
            print insertData($_GET);
            break;
        case "edit":
            // データ編集の場合
            print editData($_GET);
            break;
        case "delete":
            // データ削除の場合
            print deleteData($_GET);
            break;
        default:
            // それ以外の場合
            print ERROR;
    }
}else{
    // データが送られなかった場合
    print ERROR;
}

/**
 * 新規レコードを挿入する
 */
function insertData($data,$setid = 0){
    if(!validate($data)){
        // パラメータの妥当性を調べる
        return ERROR;
    }
    // レコードの ID を設定する
    if($setid == 0){
        $sid = getCurrentID() + 1;
    }else{
        $sid = $setid;
    }

    // ファイルを開く ( 追記 )
    $file = fopen("./data/".$data["year"].$data["month"].".dat","a");
    $stable_file = fopen("./data/id_table.dat","a");
```

```

// レコードを作製する (ID,日付,開始時間,終了時間,予約か予定)
$record = $id.", ".$data["date"].", ".$data["start"].", ".$data["finish"]."¥n";
$stable_col = $id.", ".$data["year"].$data["month"]."¥n";

// レコードを書き込む
if(fwrite($file,$record) && fwrite($stable_file,$stable_col)){
    fclose($file);
    fclose($stable_file);
}else{
    return ERROR;
}

return $id;
}

/**
 * レコードを編集する
 */
function editData($data){

    // 古いレコードを削除する
    if(!deleteData($data)){
        return ERROR;
    }

    // 新しいレコードを挿入する
    if(!insertData($data,$data["id"])){
        return ERROR;
    }

    return SUCCESS;
}

/**
 * レコードを削除する
 */
function deleteData($data){
    if(!validate($data)){
        return ERROR;
    }

    $stable = getIDTable();

    // 既に ID が設定されているかどうか調べる
    if(!array_key_exists(intval($data["id"]),$stable)){
        return ERROR;
    }

    // 予約情報ファイルからレコードを削除する
    $file = fopen("./data/".$stable[$data["id]].".dat", "r");
    $file_lines = array();
    while(!feof($file)){
        $cols = fgetcsv($file,100000,",");

```

```

        if($cols[0] != $data["id"] && $cols[0] != ""){
            array_push($file_lines,$cols);
        }
    }
    fclose($file);

    $file = fopen("./data/".$table[$data["id]].".dat","w+");
    foreach($file_lines as $line){
        fwrite($file,implode(",",$line)."\n");
    }
    fclose($file);

    // ID 表からレコードを削除する
    $file = fopen("./data/id_table.dat","w+");
    foreach($table as $key => $value){
        if($key != $data["id"] && $key != ""){
            fwrite($file,$key.", ".$value."\n");
        }
    }

    return true;
}

/**
 * 値の妥当性を調べる
 *
 * @param unknown_type $data
 * @return unknown
 */
function validate(&$data){
    if($data["mode"] != "insert"){
        if(!isset($data["id"])){
            return false;
        }
    }
    if($data["mode"] != "delete"){
        if(!isset($data["year"]) || !isset($data["month"]) || !isset($data["date"])
        || !isset($data["start"]) || !isset($data["finish"])){
            return false;
        }else{
            if(!checkdate(intval($data["month"]),intval($data["date"]),intval($data["year"])))
                return false;
        }
        if(intval($data["start"]) < START_TIME || intval($data["start"]) > FINISH_TIME ||
        intval($data["finish"]) < START_TIME || intval($data["finish"]) > FINISH_TIME){
            return false;
        }
    }
    if($data["month"] < 10 && strlen($data["month"]) < 2){
        $data["month"] = "0".$data["month"];
    }
}

```

```

    return true;
}

/**
 * ID と予約情報ファイル(年月)の対応表を取得する
 */
function getIDTable(){
    $id_table = array();
    if(file_exists("./data/id_table.dat")){
        $file = fopen("./data/id_table.dat","r");
        while(!feof($file)){
            $line = fgetcsv($file,100000,",");
            if($line[0] != ""){
                $id_table[$line[0]] = $line[1];
            }
        }
        fclose($file);
    }
    return $id_table;
}

/**
 * 現在使われている ID の数を取得する
 */
function getCurrentID(){
    $id_table = getIDTable();
    $current_key = 0;
    foreach($id_table as $key => $value){
        $key = intval($key);
        if($key > $current_key){
            $current_key = $key;
        }
    }
    return $current_key;
}
?>

```


[calendar.php]

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<style type="text/css">
<!--
.container{
    text-align:center;
}
.calendar table{
    margin:10px 20px;
    border-collapse:collapse;
}

.calendar table tr td{
    text-align:center;
    vertical-align:middle;
    border:1px solid #aaaaaa;
    font-size:0.8em;
    margin:0px;
    padding:3px;
}

.top_container{
    margin:0px auto;
    border-style:none;
    border-width:10px;
}

.top_container tr td{
    border-style:none;
    border-width:10px;
}

a:link{ color:#0000FF;text-decoration:none; }
a:hover{ color:#FFFF00;text-decoration:none; }
a:active{ color:#0000FF;text-decoration:none; }
a:visited{ color:#0000FF;text-decoration:none; }

.holiday{
    background-color:#ffcccc;
}

.saturday{
    background-color:#eeeeee;
}

.today{
    font-weight:bold;
    background-color:#ffff00;
}

.default_td{
    background-color:#f5ffff;
```

```

}

.empty_td{
    background-color:#ffffff;
}

.closed{
    color:#cccccc;
}

-->
</style>
</head>
<body>
<div class="container">

<table class="top_container">
    <tr>

<?php
require_once("../sys_config.php");
require_once("MonthlyCalendarFactory.class.php");

$year = intval(date("Y"));
$month = intval(date("m"));
$date = intval(date("j"));

for($i = 0;$i < 3;$i++){
    print "<td><div class=¥"calendar¥">";
    if($i == 0){
        $calendar = new MonthlyCalendarFactory($year,$month,$date);
    }else{
        $calendar = new MonthlyCalendarFactory($year,$month);
    }
    $calendar->draw();

    $month += 1;
    if($month > 12){
        $month -= 12;
        $year++;
    }
    print "</div></td>";
}

?>
    </tr>
</table>
</div>
</body>
</html>

```

[week.php]

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<style = "text/css">
<!--

-->
</style>
</head>
<body>
<div class="right">

<div class="container">

<h1><a href="http://for-hands.crew.sfc.keio.ac.jp/" target="_top"></a></h1>

<div class="text">
<p>左のカレンダーから予約状況を閲覧したい日付をお選びください。 </p>

<ul class="list">
  <li>* 療術を行っていない日は選択できません。 </li>
  <li>* 棒線の入った時間帯はすでに予約が入っている時間帯です。 </li>
  <li>* 1 時間半以上の空きがある時間帯から予約をお受けいたします。 </li>
</ul>
</div>

<?php
require_once("../sys_config.php");
require_once("WeeklyCalendarFactory.class.php");

// 週別予約状況表を表示する
if(isset($_GET["year"]) && isset($_GET["month"]) && isset($_GET["date"])){
    $week = new WeeklyCalendarFactory($_GET["year"],$_GET["month"],$_GET["date"]);
}else{
    $week = new WeeklyCalendarFactory(date('Y'),date('m'),date('j'));
}
$week->draw();

?>

<div class="link">
<a href="http://for-hands.crew.sfc.keio.ac.jp/" target="_top">トップページへ</a>
</div>

</div>
</div>
</body>
</html>
```

[date.php]

```
<?php
require_once("../sys_config.php");
require_once("Net/UserAgent/Mobile.php");

// 携帯のキャリアごとの表示の振り分けを行う
$agent = &Net_UserAgent_Mobile::singleton();

$format;
$font_color;
if($agent->isDocomo()){
    $format = "gif";
    $version = $agent->getHTMLVersion();
    if(isset($version)){
        if($version > 5.0){
            $font_color = "¥"#"FFFFFF¥";
        }else{
            $font_color = "¥"#"000000¥";
        }
    }else{
        $font_color = "¥"#"FFFFFF¥";
    }
}
}else if($agent->isEZweb()){
    $format = "png";
    if($agent->isXHTMLCompliant()){
        $font_color = "¥"#"FFFFFF¥";
    }else{
        $font_color = "¥"#"000000¥";
    }
}
}else if($agent->isVodafone()){
    $format = "png";
    if($agent->isType3GC() || $agent->isTypeP() || $agent->isTypeW()){
        $font_color = "¥"#"FFFFFF¥";
    }else if($agent->isTypeC()){
        $font_color = "¥"#"000000¥";
    }else{
        $font_color = "¥"#"FFFFFF¥";
    }
}
}else{
    $format = "png";
    $font_color = "¥"#"000000¥";
}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>たかくさき療術院</title>
</head>
<body>
<table>
    <tr>
        <td align="center"></td>?>
    </tr>
```

```

        <tr>
        <? print "<td bgcolor=%＼"#DD0000＼"><font color=" . $font_color ." size=%＼"1＼">今週
の予約状況</font><br></td>"; ?>
        </tr>
        <tr>
                <td align="center"><? print "<img src=%＼"image.php?year=" .
date('Y') . "&month=" . date('m') . "&date=" . date('j') . "&format=" . $format ."＼" alt=%＼"今週の予
約状況＼">"; ?>
                </td>
        </tr>
        <tr>
        <? print "<td bgcolor=%＼"#DD0000＼"><font color=" . $font_color ." size=%＼"1＼">参照
日選択</font><br></td>"; ?>
        </tr>
        <tr>
                <td><font size=1>参照したい日付を選択してください。 </font><br></td>
        </tr>

<?php
if($_GET["err"] == "nodata"){
    print "<tr><td><font color=%＼"#FF0000＼" size=%＼"1＼">日付を入力してください。
</font><br></td></tr>";
} else if($_GET["err"] == "novalid"){
    print "<tr><td><font color=%＼"#FF0000＼" size=%＼"1＼">日付の値が正しくありませ
ん。 </font><br></td></tr>";
} else if($_GET["err"] == "past"){
    print "<tr><td><font color=%＼"#FF0000＼" size=%＼"1＼">過去の日付が選択されてい
ます。 </font><br></td></tr>";
}
?>
<tr>
        <td>
                <form method="GET" action="yoyaku_joukyou.php"><?php
                print "<input type=%＼"text＼" maxlength=%＼"4＼" size=%＼"4＼"
name=%＼"year＼" value=" . date('Y') . " istyle=%＼"4＼" mode=%＼"numeric＼">年<br>%n";
                print "<input type=%＼"text＼" maxlength=%＼"2＼" size=%＼"2＼"
name=%＼"month＼" value=" . date('m') . " istyle=%＼"4＼" mode=%＼"numeric＼">月";
                print "<input type=%＼"text＼" maxlength=%＼"2＼" size=%＼"2＼"
name=%＼"date＼" value=" . date('d') . " istyle=%＼"4＼" mode=%＼"numeric＼">日<br>%n";
                ?> <input type="submit" value="参照"></form>
                </td>
        </tr>
</table>
</body>
</html>

```

1.2. UC02 : 予約状況を管理する (携帯電話用アプリケーション)

1.2.1. 試作版

[TestApp.java]

```
import com.nttdocomo.ui.*;

public class TestApp extends IApplication {

    public void start() {
        try{
            ScheduleCanvas scheduleCanvas = new ScheduleCanvas();
            Display.setCurrent(scheduleCanvas);
            scheduleCanvas.main();
        }catch(Exception e){
            e.printStackTrace();
            System.out.println("e");
        }
    }
}
```

[ScheduleCanvas.java]

```
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import java.io.*;
import javax.microedition.io.*;

public class ScheduleCanvas extends Canvas {
    private static final int D_WIDTH = Display.getWidth();

    private static final int D_HEIGHT = Display.getHeight();

    private static final int BG_COLOR = Graphics.getColorOfRGB(255, 255, 255);

    private static final int FONT_COLOR = Graphics.getColorOfRGB(0, 0, 0);

    Graphics graphics;

    Font font;

    int fontHeight;

    private static final int STATE_CALENDAR = 0;

    private static final int STATE_WEEKLY = 1;

    private static final int STATE_EDITTING = 2;

    int state = -1;

    private static final String MODE_INSERT = "insert";
```

```

private static final String MODE_EDIT = "edit";

private static final String MODE_DELETE = "delete";

String mode = MODE_INSERT;

/**
 * コンストラクタ
 */
public ScheduleCanvas() {
    font = Font.getFont(Font.SIZE_TINY);
    fontHeight = font.getHeight();
    graphics = getGraphics();
}

public void main() {
    initializeCalendar();
    processCalendar();
}

DayData today; // 今日の日付データ

DayData currentDay; // カーソルのあっている日付データ

// カレンダー初期化
void initializeCalendar() {
    today = new DayData();
    currentDay = new DayData();
}

/**
 * カレンダー画面の処理をする
 */
void processCalendar() {
    //System.out.println("1-0:" + Runtime.getRuntime().freeMemory() + "B");
    loadSchedule();
    Runtime.getRuntime().gc();
    //System.out.println("1-1:" + Runtime.getRuntime().freeMemory() + "B");
    clearKeyState();
    //System.out.println("1-2:" + Runtime.getRuntime().freeMemory() + "B");
    setSoftLabel(Frame.SOFT_KEY_1, "新規登録");
    setSoftLabel(Frame.SOFT_KEY_2, "終了");

    for (;;) {
        state = STATE_CALENDAR;
        paint(graphics);
        sleep(300);
        Runtime.getRuntime().gc();

        int key = getKeyState();
        if (key == Display.KEY_UP) {
            currentDay.move(-7);
        } else if (key == Display.KEY_DOWN) {
            currentDay.move(7);
        }
    }
}

```

```

        } else if (key == Display.KEY_LEFT) {
            currentDay.move(-1);
        } else if (key == Display.KEY_RIGHT) {
            currentDay.move(1);
        } else if (key == Display.KEY_SELECT) {
            //System.out.println("1-7:"
Runtime.getRuntime().freeMemory() + "B");
            processWeeklySchedule();
            loadSchedule();
            clearKeyState();
            setSoftLabel(Frame.SOFT_KEY_1, "新規登録");
            setSoftLabel(Frame.SOFT_KEY_2, "終了");
        } else if (key == Display.KEY_SOFT1) {
            //System.out.println("1-8:"
Runtime.getRuntime().freeMemory() + "B");
            mode = MODE_INSERT;
            processEditing();
            loadSchedule();
            clearKeyState();
            setSoftLabel(Frame.SOFT_KEY_1, "新規登録");
            setSoftLabel(Frame.SOFT_KEY_2, "終了");
            Runtime.getRuntime().gc();
        } else if (key == Display.KEY_SOFT2) {
            //System.out.println("1-9:"
Runtime.getRuntime().freeMemory() + "B");
            IApplication.getCurrentApp().terminate();
        }
    }
}

/**
 * 週別スケジュール画面の処理をする
 */
public void processWeeklySchedule() {
    loadSchedule();
    Runtime.getRuntime().gc();
    //System.out.println("2-1:" + Runtime.getRuntime().freeMemory() + "B");
    clearKeyState();
    //System.out.println("2-2:" + Runtime.getRuntime().freeMemory() + "B");
    setSoftLabel(Frame.SOFT_KEY_1, "新規登録");
    setSoftLabel(Frame.SOFT_KEY_2, "カレンダー");

    for (;;) {
        state = STATE_WEEKLY;
        paint(graphics);
        sleep(300);
        Runtime.getRuntime().gc();

        int key = getKeyState();

        if (key == Display.KEY_UP) {
            currentDay.move(-1);
        } else if (key == Display.KEY_DOWN) {
            currentDay.move(1);

```



```

        } else if (key == Display.KEY_SELECT) {
            //System.out.println("2-5:"
Runtime.getRuntime().freeMemory() + "B");
            if(numSchedule != 0){
                currentSchedule = -1;
                processDaylySchedule();
                loadSchedule();
            }
        } else if (key == Display.KEY_SOFT1) {
            //System.out.println("2-6:"
Runtime.getRuntime().freeMemory() + "B");
            mode = MODE_INSERT;
            processEditting();
            loadSchedule();
            Runtime.getRuntime().gc();
            clearKeyState();
            setSoftLabel(Frame.SOFT_KEY_1, "新規登録");
            setSoftLabel(Frame.SOFT_KEY_2, "カレンダー");
        } else if (key == Display.KEY_SOFT2) {
            //System.out.println("2-7:"
Runtime.getRuntime().freeMemory() + "B");
            //processCalendar();
            return;
        }
    }

}

/**
 * 一日のスケジュールリストの処理をする
 */
int selectCursor = -1;

public void processDaylySchedule() {
    loadSchedule();
    Runtime.getRuntime().gc();
    //System.out.println("3-1:" + Runtime.getRuntime().freeMemory() + "B");
    clearKeyState();
    //System.out.println("3-2:" + Runtime.getRuntime().freeMemory() + "B");
    setSoftLabel(Frame.SOFT_KEY_1, "新規登録");
    setSoftLabel(Frame.SOFT_KEY_2, "キャンセル");
    selectCursor = 0;

    for (;;) {
        state = STATE_WEEKLY;
        paint(graphics);
        sleep(300);
        Runtime.getRuntime().gc();

        int key = getKeyState();

        if (key == Display.KEY_UP) {
            if (selectCursor > 0) {
                selectCursor--;
            }
        }
    }
}

```

```

    }
    } else if (key == Display.KEY_DOWN) {
        if (selectCursor < numSchedule - 1) {
            selectCursor++;
        }
    } else if (key == Display.KEY_SELECT) {
        //System.out.println("3-5:"
Runtime.getRuntime().freeMemory() + "B");
        mode = MODE_EDIT;
        selectCursor = -1;
        processEditing();
        Runtime.getRuntime().gc();
        //processWeeklySchedule();
        return;
    } else if (key == Display.KEY_SOFT1) {
        //System.out.println("3-6:"
Runtime.getRuntime().freeMemory() + "B");
        mode = MODE_INSERT;
        selectCursor = -1;
        processEditing();
        Runtime.getRuntime().gc();
        //processWeeklySchedule();
        return;
    } else if (key == Display.KEY_SOFT2) {
        //System.out.println("3-7:"
Runtime.getRuntime().freeMemory() + "B");
        selectCursor = -1;
        //processWeeklySchedule();
        return;
    }
}

}

}

ScheduleData editData = new ScheduleData();

int editCursor = 0;

/**
 * スケジュール編集画面の処理をする
 */
public void processEditing() {
    editCursor = 0;
    clearKeyState();
    //System.out.println("4-1:" + Runtime.getRuntime().freeMemory() + "B");
    setSoftLabel(Frame.SOFT_KEY_1, "登録");
    setSoftLabel(Frame.SOFT_KEY_2, "中止");
    editData = new ScheduleData();
    //System.out.println("4-2:" + Runtime.getRuntime().freeMemory() + "B");
    if (mode == MODE_INSERT) {
        editData.date = currentDay;
    } else {
        editData = schedule[currentSchedule];
    }
}

```

```

for (;;) {
    state = STATE_EDITTING;
    paint(graphics);
    sleep(300);
    Runtime.getRuntime().gc();

    int key = getKeyState();

    if (key == Display.KEY_UP) {
        if (editCursor > 0) {
            editCursor--;
        }
    } else if (key == Display.KEY_DOWN) {
        if(mode == MODE_INSERT){
            if (editCursor < 10) {
                editCursor++;
            }
        } else {
            if (editCursor < 11) {
                editCursor++;
            }
        }
    } else if (key == Display.KEY_SELECT) {
        //System.out.println("4-5:"
Runtime.getRuntime().freeMemory() + "B");
        try {
            if (editCursor == 0) {
                int year =
inputNumber(editData.date.year);
                if (year >= today.year - 1 && year
<= today.year + 1) {
                    editData.date.year = year;
                    currentDay.correctDay();
                }
            } else if (editCursor == 1) {
                int month =
inputNumber(editData.date.month);
                if (month >= 1 && month <= 12) {
                    editData.date.month =
month;
                    currentDay.correctDay();
                }
            } else if (editCursor == 2) {
                int date =
inputNumber(editData.date.date);
                if (date >= 1 && date <=
DayData.getDaysInMonth(editData.date.year, editData.date.month)) {
                    editData.date.date = date;
                    currentDay.correctDay();
                }
            } else if (editCursor == 3) {
                int hour =
inputNumber(editData.startTime / 100);

```

```

100                                     if (hour >= 0 && hour < 24) {
                                           editData.startTime = hour *
                                           +
(editData.startTime % 100);
                                           }
inputNumber(editData.startTime % 100);   } else if (editCursor == 4) {
                                           int           min           =
                                           if (min >= 0 && min < 30) {
                                           editData.startTime =
                                           + 0;
                                           }else if (min >= 30 && min < 60) {
                                           editData.startTime =
                                           + 30;
                                           }
                                           } else if (editCursor == 5) {
inputNumber(editData.endTime / 100);     int           hour           =
                                           if (hour >= 0 && hour < 24) {
100                                       editData.endTime = hour *
                                           +
(editData.endTime % 100);
                                           }
                                           } else if (editCursor == 6) {
inputNumber(editData.endTime % 100);     int           min           =
                                           if (min >= 0 && min < 30) {
                                           editData.endTime =
                                           + 0;
                                           }else if (min >= 30 && min < 60) {
                                           editData.endTime =
                                           + 30;
                                           }
                                           } else if (editCursor == 7) {
inputString(editData.title,              String           title           =
                                           TextBox.KANA);
                                           if (title.length() <= 8){
                                           editData.title = title;
                                           }
                                           } else if (editCursor == 8) {
ScheduleData.RESERVE;                   editData.profile =
                                           } else if (editCursor == 9) {
                                           editData.profile =
ScheduleData.PLAN;                       } else if (editCursor >= 10){
                                           if(editCursor == 11){

```

```

mode = MODE_DELETE;
}
editData.date.correctDay();
currentDay = editData.date;
addScheduleData(editData); // 容量

オーバーのときは無視
currentSchedule = -1;
//processWeeklySchedule();
return;
}

// 開始時刻と終了時刻をチェック
if (editCursor < 2) {
    if (editData.startTime >
        editData.endTime =
    )
    } else {
        if (editData.endTime <
            editData.startTime =
        )
    }
} catch (Exception e) {
}
} else if (key == Display.KEY_SOFT1) {
    //System.out.println("4-6:"
    +
Runtime.getRuntime().freeMemory() + "B");
    editData.date.correctDay();
    currentDay = editData.date;
    addScheduleData(editData); // 容量オーバーのときは無
    視
    currentSchedule = -1;
    //processWeeklySchedule();
    return;
} else if (key == Display.KEY_SOFT2) {
    //System.out.println("4-7:"
    +
Runtime.getRuntime().freeMemory() + "B");
    currentSchedule = -1;
    return;
}
}

/**
 * 描画する
 */
public void paint(Graphics graphics) {
    if (state == -1) { //起動直後に paint が呼ばれても大丈夫にする
        return;
    }
}

```

```

graphics.lock();
graphics.setFont(font);

if (state == STATE_CALENDAR) {
    drawCalendarScreen();
} else if (state == STATE_WEEKLY) {
    drawWeeklyScreen();
} else if (state == STATE_EDITTING) {
    drawEditingScreen();
}

graphics.unlock(true);
}

/**
 * スケジュールを登録する
 *
 * @param editData
 */
public void addScheduleData(ScheduleData editData) {
    // サーバにデータを送信する
    int response = sendScheduleData(editData);
    if(mode == MODE_INSERT){
        editData.id = response;
    }
    // データを保存する
    if(response != 0){
        saveScheduleData(editData);
    }
}

/**
 * サーバにデータを送信する
 *
 * @param editData
 */
public int sendScheduleData(ScheduleData editData) {
    HttpURLConnection con = null;
    InputStream inStream = null;
    String w_remsg = "0";

    //テストサーバ
    String url = new String(IApplication.getCurrentApp().getSourceURL()
        + "alt/update.php");
    url += "?mode=" + mode;

    if(mode != MODE_DELETE){
        url += "&year=" + editData.date.year + "&month="
            + editData.date.month + "&date=" +
editData.date.date
            + "&start=" + editData.startTime + "&finish="
            + editData.endTime;
        if (editData.isReserve()) {
            url += "&profile=reserve";
        }
    }
}

```

```

        } else {
            url += "&profile=plan";
        }
    }
    if(mode != MODE_INSERT){
        url += "&id=" + editData.id;
    }

    try {
        // 接続を設定
        con = (HttpURLConnection) Connector.open(url, Connector.READ,
true);

        // 要求メソッドとコンテンツタイプを設定
        con.setRequestMethod(HttpURLConnection.GET);
        // con.setRequestProperty("Content-Type", "text/plain");

        // 実際に接続
        con.connect();

        if (con.getResponseCode() == HttpURLConnection.HTTP_OK) {
            // 入力ストリームを取得
            inStream = con.openInputStream();
            // InputStreamReader インスタンスを作成
            // SJIS を Unicode に変換
            InputStreamReader    wi_buffer    =    new
InputStreamReader(inStream);

            // StringBuffer インスタンスを作成
            StringBuffer w_StringBuffer = new StringBuffer();
            // CGI から送信されたストリームを讀取

            int w_char;
            while ((w_char = wi_buffer.read()) != -1) {
                w_StringBuffer.append((char) w_char);
            }
            w_remmsg = (String) w_StringBuffer.toString();
            // メッセージ表示領域に、CGI からの返答メッセージを設

            // w_err.setText(w_remmsg);
            //drawResult(w_remmsg);
            // 入力ストリームをクローズ
            inStream.close();
        }

        // 接続をクローズ
        con.close();

    }
    // 例外処理
    catch (Throwable e) {
        e.printStackTrace();
        // w_err.setText("書込エラー2");
    }
}

```

定

```

        return Integer.parseInt(w_remsg);
    }

    /**
     * 結果を表示する
     */
    public void drawResult(String string) {
        graphics.setColor(Graphics.getColorOfRGB(255, 255, 255));
        graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
        graphics.drawString(string, 10, 10);
        sleep(500);
    }

    /**
     * カレンダー画面を描く
     */
    void drawCalendarScreen() {
        graphics.setColor(BG_COLOR);
        graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);

        graphics.setColor(FONT_COLOR);
        graphics.drawString(currentDay.getMonthString(), 5, fontHeight + 5);

        // 書き始める日付を計算
        DayData dat = new DayData(currentDay);
        // 2週間前の日曜に戻る
        dat.move(-14);
        dat.move(-dat.day);

        // 描画開始
        int dayW = fontHeight + 6;
        int dayH = fontHeight + 3;
        int sx = (D_WIDTH - dayW * 7) / 2;
        int sy = fontHeight + 15;

        // 6週間分表示
        for (int i = 0; i < 42; i++) {
            int dx = sx + (i % 7) * dayW;
            int dy = sy + (i / 7) * dayH;
            if (i == 0 || dat.date == 1) {
                graphics.setColor(Graphics.getColorOfRGB(230, 230,
255));
                if (dat.month % 2 == 0) {
                    graphics.setColor(Graphics.getColorOfRGB(255, 230, 230));
                }
                graphics.fillRect(dx, dy, dayW * (7 - dat.day), dayH);
                if (i < 35) {
                    graphics.fillRect(sx, dy + dayH, dayW * 7,
dayH
                    * (5 - i / 7));
                }
            }
        }
    }
}

```



```

    // 今日の日付は色を変える
    if (dat.date == today.date && dat.month == today.month) {
        graphics.setColor(Graphics.getColorOfRGB(255, 255,
0));
        graphics.fillRect(dx, dy, dayW, dayH);
    }
    // 日曜と土曜は色を変える
    if (dat.day == 0) {
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    } else if (dat.day == 6) {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 255));
    } else {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }
    String str = "" + (dat.date / 10) + (dat.date % 10);
    graphics.drawString(str, dx + 2, dy + 2 + fontHeight);
    for(int k = 0; k < allplanDays.length; k++){
        if(dat.date == allplanDays[k].date && dat.month ==
allplanDays[k].month){
            graphics.drawLine(dx + 2, dy + 2 +
fontHeight/2,dx + fontHeight, dy + 2 + fontHeight/2);
        }
    }
    if (dat.date == currentDay.date && dat.month ==
currentDay.month) {
        graphics.setColor(Graphics.getColorOfRGB(0, 200, 0));
        graphics.drawRect(dx, dy, dayW - 1, dayH - 1);
        graphics.drawRect(dx + 1, dy + 1, dayW - 3, dayH - 3);
    }
    dat.move(1);
}

/**
 * for(int i = 0; i < schedule.length; i++){
 * graphics.drawString(schedule[i].title, 10, 200); }
 */

int x = 10;
int y = 120;
drawTable(x, y);
}

/**
 * 週別スケジュール画面を描く fontHeight=16(SMALL) fontHeight=12(TINY)
 */
public void drawWeeklyScreen() {
    graphics.setColor(BG_COLOR);
    graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);
    graphics.setColor(FONT_COLOR);

    graphics.drawString(currentDay.getMonthString(), 5, 5 + fontHeight);

    int x = 10;

```

```

        int y = 20;

        // スケジュール表の枠を描く
        drawTable(x, y);

        y += 125;
        // 一日のスケジュールリストを描く
        drawDaylySchedule(x, y);
    }

    /**
     * 一日のスケジュールリストを描く
     *
     */
    int numSchedule = 0;
    int currentSchedule = -1;
    public void drawDaylySchedule(int x, int y) {
        // 日付を書く
        graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
        if (currentDay.day == 0) {
            graphics.setColor(Graphics.getColorOfName(Graphics.RED));
        } else if (currentDay.day == 6) {
            graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
        }
        graphics.drawString(currentDay.getMonthDateString(), x, y);

        y += fontHeight + 3;
        graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));

        numSchedule = 0;
        for (int i = 0; i < schedule.length; i++) {
            if (schedule[i].date.compareTo(currentDay) == 0) {
                if (selectCursor == numSchedule) {

                    graphics.setColor(Graphics.getColorOfName(Graphics.RED));
                    currentSchedule = i;
                } else {

                    graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
                }
                graphics.drawString(schedule[i].getScheduleString(), x,
                    y + fontHeight * numSchedule);
                numSchedule++;
            }
        }
        // graphics.drawString("11/23(木)終日 休み", 10, 185);
    }

    /**
     * 週間スケジュールを描く
     *
     * @param x
     * @param y

```

```

*/
public void drawTable(int x, int y) {
    // スケジュール表の枠を描く
    drawTableLine(x, y);
    // スケジュール表の日付を描く
    drawTableDate(x, y);
    //
    drawTableGraph(x, y);
}

/**
 * スケジュール表を描く 横: 1 時間 = 14 ドット 縦: 一日 = 14 ドット(fontHeight+2)
 */
public void drawTableLine(int baseX, int baseY) {
    graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
    graphics.drawString("9", 48, baseY + 9);
    graphics.drawString("12", 88, baseY + 9);
    graphics.drawString("15", 130, baseY + 9);
    graphics.drawString("18", 172, baseY + 9);
    graphics.drawString("21", 214, baseY + 9);

    int originX = baseX + 40;
    int originY = baseY + fontHeight;
    int hourW = 14;
    int dayH = fontHeight+1;

    // 横線を引く
    for (int i = 0; i < 8; i++) {
        int height = i * dayH;
        graphics.drawLine(originX, originY + height, originX + 168,
originY
                                + height);
    }

    // 1 時間区切りの線を描く
    for (int i = 0; i < 13; i++) {
        int width = i * hourW;
        graphics.drawLine(originX + width, originY - 2, originX + width,
                                originY + dayH*7);
    }

    // 30 分区切りの線を描く
    // originX += hourW/2;
    for (int i = 0; i < 12; i++) {
        int width = i * hourW;
        drawVerticalDottedLine(originX + width + hourW / 2, originY,
                                originX + width + hourW / 2, originY +
dayH*7);
    }

    // 選択された日に枠を描く
    graphics.setColor(Graphics.getColorOfRGB(0, 200, 0));
    int currentH = dayH * currentDay.day;
    graphics.drawRect(originX, originY + currentH, 168, dayH);
}

```

```

        graphics.drawRect(originX - 1, originY + currentH - 1, 168 + 2,
                           dayH + 2);
    }

    /**
     * 日にちと曜日を描く
     */
    public void drawTableDate(int x, int y) {
        // 書き始める日付を計算
        DayData dat = new DayData(currentDay);
        // 日曜に戻る
        dat.move(-dat.day);

        y += fontHeight;
        // 日曜を描く
        graphics.setColor(Graphics.getColorOfName(Graphics.RED));
        graphics.drawString(dat.getDateString(), x, y + fontHeight);
        dat.move(1);
        y += fontHeight+1;

        // 平日を描く
        graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
        for (int i = 0; i < 5; i++) {
            graphics.drawString(dat.getDateString(), x, y + fontHeight);
            dat.move(1);
            y += fontHeight+1;
        }

        // 土曜を描く
        graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
        graphics.drawString(dat.getDateString(), x, y + fontHeight);
    }

    /**
     * 棒線を描く
     */
    private void drawTableGraph(int x, int y) {
        int originX = x + 40;
        int originY = y + fontHeight;

        // スケジュールの時間を表す棒を描く
        graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));

        // 書き始める日付、書き終わる日付を計算
        DayData startDate = new DayData(currentDay);
        DayData endDate = new DayData(currentDay);
        startDate.move(-startDate.day);
        endDate.move(6 - endDate.day);

        for (int i = 0; i < schedule.length; i++) {
            if (schedule[i].date.compareTo(startDate) >= 0
                && schedule[i].date.compareTo(endDate) <=
0) {

```

```

        if(schedule[i].profile == ScheduleData.RESERVE){
graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
        }else{
graphics.setColor(Graphics.getColorOfName(Graphics.GRAY));
        }
graphics.fillRect(originX
transform(schedule[i].startTime),
originY + 4 + 13
schedule[i].date.day,
transform(schedule[i].endTime)
transform(schedule[i].startTime), 6);
    }
}

public int transform(int time) {
    int point = (time - 900) / 100 * 14;
    if ((time - 900) % 100 != 0) {
        point += 8;
    }
    return point;
}

/**
 * スケジュール編集画面を描く
 */
public void drawEditingScreen() {
    graphics.setColor(Graphics.getColorOfRGB(255, 255, 255));
    graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);

    drawTable(10, 10);

    int x = 20;
    int y = 120;

    // 日付を描画する
    graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    graphics.drawString("日付", x, y + fontHeight + 3);

    x += fontHeight * 2 + 3;
    drawBox(editData.date.getYear(), x, y, 4, 1, editCursor == 0);

    x += fontHeight * 2 + 8;
    graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    graphics.drawString("年", x, y + fontHeight + 3);

    //y += fontHeight + 3;

    x += fontHeight;
    drawBox(editData.date.getMonth(), x, y, 2, 1, editCursor == 1);
}

```

```

x += fontHeight + 8;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("月", x, y + fontHeight + 3);

x += fontHeight;
drawBox(editData.date.getDate(), x, y, 2, 1, editCursor == 2);

x += fontHeight + 8;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("日", x, y + fontHeight + 3);

// 予定時刻の入力フォームを描画をする
y += fontHeight + 8;
x = 20;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("時間", x, y + fontHeight + 3);

// 開始時刻
//y += fontHeight + 3;
x += fontHeight * 2 + 3;
drawBox(editData.getStartHour(), x, y, 2, 1, editCursor == 3);

x += fontHeight + 8;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString(":", x, y + fontHeight + 3);

x += fontHeight / 2;
drawBox(editData.getStartMinute(), x, y, 2, 1, editCursor == 4);

x += fontHeight + 12;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("-", x, y + fontHeight + 3);

// 終了時刻
x += fontHeight / 2 + 4;
drawBox(editData.getEndHour(), x, y, 2, 1, editCursor == 5);

x += fontHeight + 8;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString(":", x, y + fontHeight + 3);

x += fontHeight / 2;
drawBox(editData.getEndMinute(), x, y, 2, 1, editCursor == 6);

// 予定の名前の入力フォームを描画する
x = 20;
int c = 16;
//(D_WIDTH - 10) / (fontHeight / 2);

y += fontHeight + 8;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("名前", x, y + fontHeight + 3);

x += fontHeight * 2 + 3;

```

```

drawBox(editData.title, x, y, c, 1, editCursor == 7);

// 予定の種類ラジオボタンを描画する
x = 20;
y += fontHeight + 8;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("予定の種類", x, y + fontHeight);

y += fontHeight + 5;
drawRadio(editData.profile == ScheduleData.RESERVE, x, y, editCursor
== 8);

x += fontHeight;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("予約", x, y + fontHeight / 2 + 3);

//x = 20;
//y += fontHeight + 3;
x += fontHeight * 3;
drawRadio(editData.profile == ScheduleData.PLAN, x, y, editCursor ==
9);

x += fontHeight;
graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
graphics.drawString("用事", x, y + fontHeight / 2 + 3);

y += fontHeight + 5;
x = 20;
if(mode == MODE_INSERT){
    drawButton("登録", x, y, editCursor == 10);
    return;
}else{
    String str = "変更";
    drawButton(str, x, y, editCursor == 10);
    x += (str.length() + 1) * fontHeight;
    drawButton("削除", x, y, editCursor == 11);
}
}

/**
 * テキストフォームを描く
 */
void drawBox(String str, int x, int y, int col, int row, boolean focus) {
    int w = col * fontHeight / 2 + 6;
    int h = row * (fontHeight + 1) + 3;

    if (focus) {
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    } else {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }
    graphics.drawLine(x, y + 1, x, y + h - 1);
    graphics.drawLine(x + w, y + 1, x + w, y + h - 1);
    graphics.drawLine(x + 1, y, x + w - 1, y);
    graphics.drawLine(x + 1, y + h, x + w - 1, y + h);
}

```

```

graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));

int sy = y + fontHeight + 3;
int idx = 0;
while (sy < y + h && idx < str.length()) {
    int idx2 = font.getLineBreak(str, 0, str.length(), w - 6);
    int idx3 = str.indexOf('\n', idx);
    if (idx3 >= 0 && idx3 < idx2) {
        idx2 = idx3 + 1;
    }
    graphics.drawString(str.substring(idx, idx2), x + 3, sy);
    sy += fontHeight + 1;
    idx = idx2;
}
}

/**
 * ラジオボタンを描く
 */
void drawRadio(boolean chosen, int x, int y, boolean focus) {
    // 枠を描く
    if (focus) {
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    } else {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }
    graphics.drawArc(x, y, fontHeight - 4, fontHeight - 4, 0, 360);

    // 中の点を描く
    graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    if (chosen) {
        graphics.fillArc(x + 2, y + 2, (fontHeight - 2) / 2,
            (fontHeight - 2) / 2, 0, 360);
    }
}

/**
 * ボタンを描く
 */
void drawButton(String str, int x, int y, boolean focus){
    if(focus){
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    }else{
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }

    int width = str.length() * fontHeight;
    graphics.drawRect(x, y, width + 4, fontHeight + 4);
    graphics.drawString(str, x + 3, y + 2 + fontHeight);
}

/**
 * 縦の点線を描く
 */

```



```

    * @param x1
    * @param y1
    * @param x2
    * @param y2
    */
    public void drawVerticalDottedLine(int x1, int y1, int x2, int y2) {
        graphics.setColor(Graphics.getColorOfName(Graphics.GRAY));
        for (int i = 0; i < y2 - y1; i += 4) {
            int j = i + 2;
            graphics.drawLine(x1, y1 + i, x2, y1 + j);
        }
    }
}

/*****
 * スクラッチパッド
 *****/

private static final int SP_SCHEDULE_DATA = 0;
private static final int SP_SCHEDULE_DATA_LENGTH = 10240;

ScheduleData[] schedule;
DayData[] allplanDays;
/**
 * データを読み込む
 */
void loadSchedule() {
    try {
        ByteArrayInputStream is = new ByteArrayInputStream(load(
            SP_SCHEDULE_DATA_LENGTH));
        int num = (is.read() << 8) | is.read();
        schedule = new ScheduleData[num];
        for (int i = 0; i < schedule.length; i++) {
            schedule[i] = ScheduleData.read(is);
        }
        is.close();
    } catch (Throwable e) {
        System.out.println("" + e);
    }

    int counter = 0;
    for(int i = 0; i < schedule.length; i++){
        if(schedule[i].isAlldayPlan()){
            counter++;
        }
    }
    allplanDays = new DayData[counter];

    int k = 0;
    for(int i = 0; i < schedule.length; i++){
        if(schedule[i].isAlldayPlan()){
            allplanDays[k] = schedule[i].date;
            k++;
        }
    }
}

```

```

    }
}

/**
 * データを保存する
 *
 * @param string
 */
public boolean saveScheduleData(ScheduleData data) {
    deleteScheduleData(data);

    if(mode != MODE_DELETE){
        ScheduleData[] dummy = schedule;
        schedule = new ScheduleData[dummy.length + 1];
        int n = 0;
        boolean added = false;
        for (int i = 0; i < dummy.length; i++) {
            if (!added) {
                int def = dummy[i].date.compareTo(data.date);
                if (def > 0 || def == 0 && dummy[i].startTime
> data.startTime) {
                    schedule[n++] = data;
                    added = true;
                }
            }
            schedule[n++] = dummy[i];
        }
        if (!added) { // どのデータよりも時間が大きい場合はここ
            schedule[n++] = data;
        }
    }

    return saveToScratchpad();
}

// スケジュールデータの削除
void deleteScheduleData(ScheduleData data) {
    int idx = -1;
    for (int i = 0; i < schedule.length; i++) {
        if (schedule[i] == data) {
            idx = i;
            break;
        }
    }
    if (idx < 0) {
        return;
    }
    ScheduleData[] dummy = schedule;
    schedule = new ScheduleData[dummy.length - 1];
    int n = 0;
    for (int i = 0; i < dummy.length; i++) {
        if (i != idx) {
            schedule[n++] = dummy[i];
        }
    }
}

```

```

    }
}

boolean saveToScratchpad() {
    try {
        ByteArrayOutputStream os = new ByteArrayOutputStream();
        os.write(schedule.length >> 8);
        os.write(schedule.length);
        for (int i = 0; i < schedule.length; i++) {
            schedule[i].write(os);
        }
        byte[] b = os.toByteArray();
        os.close();

        if (b.length > SP_SCHEDULE_DATA_LENGTH) {
            return false;
        }

        save(b, SP_SCHEDULE_DATA);
    } catch (Throwable e) {
    }
    return false;
}

// スクラッチパッドへの書き込み
void save(byte[] dat, int pos) {
    OutputStream os = null;
    try {
        os = Connector.openOutputStream("scratchpad:///0;pos=" + pos);
        os.write(dat);
    } catch (Throwable e) {
    }
    try {
        os.close();
    } catch (Throwable e) {
    }
}

// スクラッチパッドからの読み込み
byte[] load(int pos, int len) {
    InputStream is = null;
    byte[] ret = new byte[len];
    try {
        is = Connector.openInputStream("scratchpad:///0;pos=" + pos);
        is.read(ret);
    } catch (Throwable e) {
        ret = null;
    }
    try {
        is.close();
    } catch (Throwable e) {
    }
    return ret;
}

```

```

}

/** *****
*/

// イベント*****
int keyParam = -1;

public void processEvent(int type, int param) {
    if (type == Display.KEY_PRESSED_EVENT) {
        keyParam = param;
    }
}

// キー状態の取得
int getKeyState() {
    int ret = keyParam;
    keyParam = -1;
    return ret;
}

// キー状態のクリア
void clearKeyState() {
    keyParam = -1;
}

// その他処理*****
void sleep(int time) {
    try {
        Thread.sleep(time);
    } catch (Throwable e) {
    }
}

boolean waitIME = false;

String imeString = null;

/**
 * 入力された数字を取得する
 *
 * @param value
 * @return
 */
int inputNumber(int value) {
    try {
        String str = inputString(Integer.toString(value), "" + (value / 10)
+ (value % 10),
                TextBox.NUMBER);
        return Integer.parseInt(str);
    } catch (Exception e) {
    }
    return value;
}

```

```

/**
 * 入力された文字を取得する
 *
 * @param str
 * @param mode
 * @return
 */
String inputString(String str, int mode) {
    while (getKeypadState() > 0) {
        sleep(100);
    }

    waitIME = true;
    imeOn(str, TextBox.DISPLAY_ANY, mode);

    while (waitIME) {
        sleep(100);
    }

    while (getKeypadState() > 0) {
        sleep(100);
    }
    return imeString;
}

public void processIMEEvent(int type, String str) {
    imeString = str;
    waitIME = false;
}
}

```

[DayData.java]

```

import java.util.Calendar;
import java.util.Date;

public class DayData {
    int year; // 年

    int month; // 月 : 1月が 1

    int date; // 日

    int day; // 曜日 : 日曜が 0

    DayData() {
        this(System.currentTimeMillis());
    }

    DayData(long time) {
        Calendar cal = Calendar.getInstance();
        cal.setTime(new Date(time));
        year = cal.get(Calendar.YEAR);
    }
}

```

```

        month = cal.get(Calendar.MONTH) + 1;
        date = cal.get(Calendar.DAY_OF_MONTH);
        day = cal.get(Calendar.DAY_OF_WEEK) - Calendar.SUNDAY;
    }

```

```

DayData(DayData d) {
    this.year = d.year;
    this.month = d.month;
    this.date = d.date;
    this.day = d.day;
}

```

//曜日を補正する

```

void correctDay(){
    Calendar cal = Calendar.getInstance();
    cal.set(Calendar.YEAR, this.year);
    cal.set(Calendar.MONTH, this.month-1);
    cal.set(Calendar.DATE, this.date);
    day = cal.get(Calendar.DAY_OF_WEEK) - Calendar.SUNDAY;
}

```

// 日付を引数だけ進める。引数がマイナスの場合戻る

```

void move(int def) {
    while (def > 0) {
        date++;
        if (date > getDaysInMonth(year, month)) {
            date = 1;
            month++;
            if (month > 12) {
                month = 1;
                year++;
            }
        }
        day = (day + 1) % 7;
        def--;
    }
    while (def < 0) {
        date--;
        if (date < 1) {
            month--;
            if (month < 1) {
                month = 12;
                year--;
            }
            date = getDaysInMonth(year, month);
        }
        day = (day + 6) % 7;
        def++;
    }
}

```

// 日付の比較。dのほうが大きい場合はマイナス。 this-d と覚えれば OK

```

int compareTo(DayData d) {
    int prm1 = this.year * 10000 + this.month * 100 + this.date;

```

```

        int prm2 = d.year * 10000 + d.month * 100 + d.date;
        return prm1 - prm2;
    }

    // 年と月から、その月の日数を計算する
    static final int[] DAYS_IN_MONTH = { // それぞれの月の日数
        31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, };

    static int getDaysInMonth(int y, int m) {
        if (m == 2 && y % 4 == 0) { // 100年に一度とか、2000年に一度は無視
            return 29;
        }
        return DAYS_IN_MONTH[m - 1];
    }

    // 文字列に変換
    String getMonthString() {
        return "" + year + "年" + (month / 10) + (month % 10) + "月";
    }

    // 月と曜日を文字列に変換する
    String getMonthDateString() {
        return month + "月" + date + "(" + getKanjiOf(day) + ")";
    }

    // 日と曜日を文字列に変換する
    String getDateString() {
        String preposition = "";
        if(date < 10){
            preposition = " ";
        }
        return preposition + date + "(" + getKanjiOf(day) + ")";
    }

    // 曜日を漢字に変換する
    String getKanjiOf(int day){
        switch (day){
            case 0:
                return "日";
            case 1:
                return "月";
            case 2:
                return "火";
            case 3:
                return "水";
            case 4:
                return "木";
            case 5:
                return "金";
            case 6:
                return "土";
            default:
                return "";
        }
    }

```

```

    }

    public String getYear() {
        return Integer.toString(year);
    }

    public String getDate() {
        return "" + (date / 10) + (date % 10);
    }

    public String getMonth() {
        return "" + (month / 10) + (month % 10);
    }
}

```

[ScheduleData.java]

```

import java.io.InputStream;
import java.io.OutputStream;

public class ScheduleData {
    int id;
    DayData date;           //日付
    int startTime, endTime; //開始時刻、終了時刻。HHMM と入る
    String title;          //タイトル
    boolean profile;      //予定の内容

    //定数
    static final boolean RESERVE = true; //予約
    static final boolean PLAN = false;  //用事

    //コンストラクタ
    ScheduleData() {
        date = new DayData();
        startTime = 900;
        endTime = 1800;
        title = "";
        profile = RESERVE;
    }

    //データの書き込み
    void write(OutputStream os) throws Exception {
        os.write(date.year >> 8);
        os.write(date.year);
        os.write(date.month);
        os.write(date.date);
        os.write(date.day);

        os.write(startTime >> 8);
        os.write(startTime);
        os.write(endTime >> 8);
    }
}

```



```

        os.write(endTime);

        if(profile){
            os.write(1);
        }else{
            os.write(0);
        }

        os.write(id >> 24);
        os.write(id >> 16);
        os.write(id >> 8);
        os.write(id);

        byte[] b = title.getBytes();
        os.write(b.length >> 8);
        os.write(b.length);
        os.write(b);
    }

//データの読み込み
static ScheduleData read(InputStream is) throws Exception {
    ScheduleData sd = new ScheduleData();
    sd.date.year = (is.read() << 8) | is.read();
    sd.date.month = is.read();
    sd.date.date = is.read();
    sd.date.day = is.read();

    sd.startTime = (is.read() << 8) | is.read();
    sd.endTime = (is.read() << 8) | is.read();

    if(is.read() == 1){
        sd.profile = true;
    }else{
        sd.profile = false;
    }

    sd.id = (is.read() << 24) | (is.read() << 16) | (is.read() << 8) | is.read();

    int len = (is.read() << 8) | is.read();
    byte[] b = new byte[len];
    is.read(b);
    sd.title = new String(b);

    return sd;
}

public String getScheduleString(){
    return getStartHour() + ":" + getStartMinute() + " " + title;
}

String getStartHour() {
    int h = startTime / 100;
    return "" + (h / 10) + (h % 10);
}

```

```

    }

    String getStartMinute() {
        int m = startTime % 100;
        return "" + (m / 10) + (m % 10);
    }

    String getEndHour() {
        int h = endTime / 100;
        return "" + (h / 10) + (h % 10);
    }

    String getEndMinute() {
        int m = endTime % 100;
        return "" + (m / 10) + (m % 10);
    }

    boolean isReserve(){
        return profile;
    }

    boolean isAlldayPlan(){
        if(startTime == 900 && endTime == 2100 && profile == PLAN){
            return true;
        }else{
            return false;
        }
    }
}

```

1.2.2. 製品版

[ForHandsApp.java]

```

import com.nttdocomo.ui.*;

public class ForHandsApp extends IApplication {

    public void start() {
        try{
            ScheduleCanvas scheduleCanvas = new ScheduleCanvas();
            PhoneSystem.setAttribute(PhoneSystem.DEV_KEYPAD, 1);
            Display.setCurrent(scheduleCanvas);
            scheduleCanvas.main();
        }catch(Exception e){
            e.printStackTrace();
            System.out.println("e");
        }
    }
}

```

[ScheduleCanvas.java]

```

import com.nttdocomo.ui.*;

```

```

import com.nttdocomo.io.*;
import java.io.*;
import javax.microedition.io.*;

public class ScheduleCanvas extends Canvas {
    private static final int D_WIDTH = Display.getWidth();

    private static final int D_HEIGHT = Display.getHeight();

    private static final int BG_COLOR = Graphics.getColorOfRGB(255, 255, 255);

    private static final int FONT_COLOR = Graphics.getColorOfRGB(0, 0, 0);

    Graphics graphics;

    Font font;

    int fontHeight;

    private static final int STATE_CALENDAR = 0;

    private static final int STATE_SELECT_START = 1;

    private static final int STATE_SELECT_END = 2;

    private static final int STATE_EDITTING = 3;

    int state = -1;

    private static final String MODE_INSERT = "insert";

    private static final String MODE_EDIT = "edit";

    private static final String MODE_DELETE = "delete";

    String mode = MODE_INSERT;

    private static final int WAIT_TIME = 300;

    int editCursor = 0;

    DayData today; // 今日の日付データ

    DayData currentDay; // カーソルのあっている日付データ

    ScheduleData editData;

    ScheduleData selectedSchedule;

    boolean cursorState = false;

    private static final boolean ON_SCHEDULE = true;

    private static final boolean OFF_SCHEDULE = false;

```

```

/**
 * コンストラクタ
 */
public ScheduleCanvas() {
    font = Font.getFont(Font.SIZE_TINY);
    fontHeight = font.getHeight();
    graphics = getGraphics();
}

public void main() {
    initializeCalendar();
    processCalendar();
}

// カレンダー初期化
void initializeCalendar() {
    today = new DayData();
    currentDay = new DayData();
    editData = new ScheduleData();
    selectedSchedule = new ScheduleData();
}

/**
 * カレンダー画面の処理をする
 */
void processCalendar() {
    loadSchedule();
    Runtime.getRuntime().gc();
    clearKeyState();
    setSoftLabel(Frame.SOFT_KEY_1, "決定");
    setSoftLabel(Frame.SOFT_KEY_2, "終了");

    for (;;) {
        state = STATE_CALENDAR;
        paint(graphics);
        sleep(WAIT_TIME);
        Runtime.getRuntime().gc();

        int key = getKeyState();
        int keypad = getKeyState();
        if (key == Display.KEY_UP || (keypad & (1 << Display.KEY_UP)) > 0) {
            currentDay.move(-7);
        } else if (key == Display.KEY_DOWN
            || (keypad & (1 << Display.KEY_DOWN)) > 0) {
            currentDay.move(7);
        } else if (key == Display.KEY_LEFT
            || (keypad & (1 << Display.KEY_LEFT)) > 0) {
            currentDay.move(-1);
        } else if (key == Display.KEY_RIGHT
            || (keypad & (1 << Display.KEY_RIGHT)) > 0) {
            currentDay.move(1);
        } else if (key == Display.KEY_SELECT || key == Display.KEY_SOFT1) {
            processSelectStart();
        }
    }
}

```

```

        loadSchedule();
        clearKeyState();
        setSoftLabel(Frame.SOFT_KEY_1, "決定");
        setSoftLabel(Frame.SOFT_KEY_2, "終了");
    } else if (key == Display.KEY_SOFT2) {
        IApplication.getCurrentApp().terminate();
    }
}

/**
 * 週別スケジュール画面の処理をする 開始時間を選ぶ
 */
public void processSelectStart() {
    loadSchedule();
    Runtime.getRuntime().gc();
    clearKeyState();
    setSoftLabel(Frame.SOFT_KEY_1, "決定");
    setSoftLabel(Frame.SOFT_KEY_2, "戻る");

    for (;;) {
        state = STATE_SELECT_START;
        paint(graphics);
        sleep(WAIT_TIME);
        Runtime.getRuntime().gc();

        int key = getKeyState();
        int keypad = getKeypadState();

        if (key == Display.KEY_UP || (keypad & (1 << Display.KEY_UP)) > 0) {
            if (cursorState == ON_SCHEDULE) {
                cursorState = OFF_SCHEDULE;
            }
            currentDay.move(-1);
        } else if (key == Display.KEY_DOWN
            || (keypad & (1 << Display.KEY_DOWN)) > 0) {
            if (cursorState == ON_SCHEDULE) {
                cursorState = OFF_SCHEDULE;
            }
            currentDay.move(1);
        } else if (key == Display.KEY_LEFT
            || (keypad & (1 << Display.KEY_LEFT)) > 0) {
            // 開始時間を 30 分前にずらす
            if (cursorState == ON_SCHEDULE) {
                editData.startTime = selectedSchedule.startTime;
                cursorState = OFF_SCHEDULE;
            }
            editData.shiftStartTime(-1);
        } else if (key == Display.KEY_RIGHT
            || (keypad & (1 << Display.KEY_RIGHT)) > 0) {
            // 開始時間を 30 分後ろにずらす
            if (cursorState == ON_SCHEDULE) {
                editData.setStartTime(selectedSchedule.endTime);
                cursorState = OFF_SCHEDULE;
            }
        }
    }
}

```

```

    } else {
        editData.shiftStartTime(1);
    }
} else if (key == Display.KEY_SELECT || key == Display.KEY_SOFT1) {
    if (cursorState == ON_SCHEDULE) { // 予定があったら編集する
        mode = MODE_EDIT;
        processEdit();
    } else { // 開始時間を決定して終了時間を計算する
        if (editData.startTime == 2230) {
            editData.setEndTime(editData.startTime + 70);
        } else if (isInside(editData.startTime + 70)) {
            if (editData.startTime % 100 == 0) {
                editData.setEndTime(editData.startTime + 30);
            } else {
                editData.setEndTime(editData.startTime + 70);
            }
        } else {
            editData.setEndTime(editData.startTime + 100);
        }
        processSelectEnd();
    }
    loadSchedule();
    editData.title = "";
} else if (key == Display.KEY_SOFT2 || key == Display.KEY_CLEAR) {
    return;
} else if (key == Display.KEY_CLEAR) {
    return;
}
}
}

/**
 * 終了時間を選ぶ
 */
public void processSelectEnd() {
    loadSchedule();
    Runtime.getRuntime().gc();

    for (;;) {
        state = STATE_SELECT_END;
        paint(graphics);
        sleep(WAIT_TIME);
        Runtime.getRuntime().gc();

        int key = getKeyState();
        int keypad = getKeypadState();
        if (key == Display.KEY_LEFT
            || (keypad & (1 << Display.KEY_LEFT)) > 0) {
            // 終了時間を 30 分前にずらす
            editData.shiftEndTime(-1);
        } else if (key == Display.KEY_RIGHT
            || (keypad & (1 << Display.KEY_RIGHT)) > 0) {
            // 終了時間を 30 分後ろにずらす
            if (!isInside(editData.endTime)) {

```

```

        editData.shiftEndTime(1);
    }
} else if (key == Display.KEY_SELECT || key == Display.KEY_SOFT1) {
    // 終了時間を決定する
    mode = MODE_INSERT;
    processEdit();
    return;
} else if (key == Display.KEY_SOFT2 || key == Display.KEY_CLEAR) {
    return;
}
}
}

/**
 * 【】が他の予約時間の範囲内か判定する
 *
 * @return
 */
public boolean isInside(int endTime) {
    boolean flag = false;
    for (int i = 0; i < schedule.length; i++) {
        if (currentDay.compareTo(schedule[i].date) == 0
            && endTime >= schedule[i].startTime
            && endTime <= schedule[i].endTime) {
            flag = true;
        }
    }
    return flag;
}

/**
 * スケジュール編集画面の処理をする
 */
public void processEdit() {
    editCursor = 0;

    if (mode == MODE_INSERT) {
        editData.date = currentDay;
    } else {
        editData = selectedSchedule;
    }

    for (;;) {
        state = STATE_EDITTING;
        paint(graphics);
        sleep(WAIT_TIME);
        Runtime.getRuntime().gc();

        int key = getKeyState();

        if (key == Display.KEY_UP) {
            if (editCursor > 0) {
                editCursor--;
            }
        }
    }
}

```

```

    } else if (key == Display.KEY_DOWN) {
        if (mode == MODE_INSERT) {
            if (editCursor < 1) {
                editCursor++;
            }
        } else {
            if (editCursor < 2) {
                editCursor++;
            }
        }
    } else if (key == Display.KEY_SELECT || key == Display.KEY_SOFT1) {
        try {
            if (editCursor == 0) {
                String title = inputString(editData.title, TextBox.KANA);
                if (title.length() <= 6) {
                    editData.title = title;
                }
            } else if (editCursor >= 1) {
                if (editCursor == 2) {
                    mode = MODE_DELETE;
                }
                // 予約を登録する
                addScheduleData(editData);
                return;
            }
        } catch (Exception e) {
        }
    } else if (key == Display.KEY_SOFT2 || key == Display.KEY_CLEAR) {
        return;
    }
}

}

/**
 * スケジュールを登録する
 *
 * @param editData
 */
public void addScheduleData(ScheduleData editData) {
    // サーバにデータを送信する
    if (mode != MODE_EDIT) {
        int response = sendScheduleData(editData);
        if (mode == MODE_INSERT) {
            editData.id = response;
        }
        // データを保存する
        if (response != 0) {
            saveScheduleData(editData);
        }
    } else {
        saveScheduleData(editData);
    }
}
}

```



```

/**
 * サーバにデータを送信する
 *
 * @param editData
 */
public int sendScheduleData(ScheduleData editData) {
    HttpURLConnection con = null;
    InputStream inStream = null;
    String w_remsg = "0";

    // データを送信するサーバ
    String url = new String(IApplication.getCurrentApp().getSourceURL()
        + "reserve/update.php");
    url += "?mode=" + mode;

    if (mode == MODE_INSERT) {
        url += "&year=" + editData.date.year + "&month="
            + editData.date.month + "&date=" + editData.date.date
            + "&start=" + editData.startTime + "&finish="
            + editData.endTime;
    }
    if (mode == MODE_DELETE) {
        url += "&id=" + editData.id;
    }

    try {
        // 接続を設定
        con = (HttpURLConnection) Connector.open(url, Connector.READ, true);

        // 要求メソッドとコンテンツタイプを設定
        con.setRequestMethod(HttpURLConnection.GET);

        // 実際に接続
        con.connect();

        if (con.getResponseCode() == HttpURLConnection.HTTP_OK) {
            // 入力ストリームを取得
            inStream = con.openInputStream();
            // InputStreamReader インスタンスを作成
            // SJIS を Unicode に変換
            InputStreamReader wi_buffer = new InputStreamReader(inStream);
            // StringBuffer インスタンスを作成
            StringBuffer w_StringBuffer = new StringBuffer();
            // CGI から送信されたストリームを読取

            int w_char;
            while ((w_char = wi_buffer.read()) != -1) {
                w_StringBuffer.append((char) w_char);
            }
            w_remsg = (String) w_StringBuffer.toString();

            // 入力ストリームをクローズ
            inStream.close();

```

```

    }

    // 接続をクローズ
    con.close();

}
// 例外処理
catch (Throwable e) {
    e.printStackTrace();
}
return Integer.parseInt(w_remsg);
}

/**
 * 描画する
 */
public void paint(Graphics graphics) {
    if (state == -1) { // 起動直後に paint が呼ばれても大丈夫にする
        return;
    }
    graphics.lock();
    graphics.setFont(font);

    if (state == STATE_CALENDAR) {
        drawCalendar();
    } else if (state == STATE_SELECT_START || state == STATE_SELECT_END) {
        drawWeekSchedule();
    } else if (state == STATE_EDITTING) {
        drawEdit();
    }

    graphics.unlock(true);
}

/**
 * カレンダー画面を描く
 */
void drawCalendar() {
    graphics.setColor(BG_COLOR);
    graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);

    graphics.setColor(FONT_COLOR);
    graphics.drawString(currentDay.getMonthString(), 5, fontHeight + 5);

    // 書き始める日付を計算
    DayData dat = new DayData(currentDay);
    // 2週間前の日曜に戻る
    dat.move(-14);
    dat.move(-dat.day);

    // 描画開始
    int dayW = fontHeight + 6;
    int dayH = fontHeight + 3;
    int sx = (D_WIDTH - dayW * 7) / 2;

```

```

int sy = fontHeight + 15;

// 6週間分表示
for (int i = 0; i < 42; i++) {
    int dx = sx + (i % 7) * dayW;
    int dy = sy + (i / 7) * dayH;
    if (i == 0 || dat.date == 1) {
        graphics.setColor(Graphics.getColorOfRGB(230, 230, 255));
        if (dat.month % 2 == 0) {
            graphics.setColor(Graphics.getColorOfRGB(255, 230, 230));
        }
        graphics.fillRect(dx, dy, dayW * (7 - dat.day), dayH);
        if (i < 35) {
            graphics.fillRect(sx, dy + dayH, dayW * 7, dayH
                * (5 - i / 7));
        }
    }

    // 今日の日付は色を変える
    if (dat.date == today.date && dat.month == today.month) {
        graphics.setColor(Graphics.getColorOfRGB(255, 255, 0));
        graphics.fillRect(dx, dy, dayW, dayH);
    }

    // 日曜と土曜は色を変える
    if (dat.day == 0) {
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    } else if (dat.day == 6) {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 255));
    } else {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }
    String str = "" + (dat.date / 10) + (dat.date % 10);
    graphics.drawString(str, dx + 2, dy + 2 + fontHeight);

    // 終日の予定を入れたら日にちに横線を描く
    for (int k = 0; k < allDaySchedules.length; k++) {
        if (dat.compareTo(allDaySchedules[k]) == 0) {
            graphics.drawLine(dx + 2, dy + 2 + fontHeight / 2, dx
                + fontHeight, dy + 2 + fontHeight / 2);
            break;
        }
    }

    // 選択中の日に枠を描く
    if (dat.date == currentDay.date && dat.month == currentDay.month) {
        graphics.setColor(Graphics.getColorOfRGB(0, 200, 0));
        graphics.drawRect(dx, dy, dayW - 1, dayH - 1);
        graphics.drawRect(dx + 1, dy + 1, dayW - 3, dayH - 3);
    }
    dat.move(1);
}

int x = 1;

```

```

    int y = 120;
    // スケジュール表を描く
    drawTable(x, y);
}

/**
 * 週別スケジュール画面を描く
 */
public void drawWeekSchedule() {
    graphics.setColor(BG_COLOR);
    graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);

    graphics.setColor(FONT_COLOR);
    graphics.drawString(currentDay.getMonthString(), 5, 5 + fontHeight);

    graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
    if (state == STATE_SELECT_START) {
        graphics.drawString("開始時間を選択してください", 76, 5 + fontHeight);
    } else if (state == STATE_SELECT_END) {
        graphics.drawString("終了時間を選択してください", 76, 5 + fontHeight);
    }

    int x = 1;
    int y = 20;

    // スケジュール表を描く
    drawTable(x, y);

    x = 10;
    y += 118;
    // 一日のスケジュールリストを描く
    drawDailySchedule(x, y);
}

/**
 * 一日のスケジュールリストを描く
 */
public void drawDailySchedule(int x, int y) {
    // 日付を書く
    graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
    if (currentDay.day == 0) {
        graphics.setColor(Graphics.getColorOfName(Graphics.RED));
    } else if (currentDay.day == 6) {
        graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
    }
    graphics.drawString(currentDay.getMonthDateString(), x, y);

    // 時間を書く
    if (cursorState == ON_SCHEDULE) {
        graphics.drawString(selectedSchedule.getStartHour() + ":"
            + selectedSchedule.getStartMinute() + " - "
            + selectedSchedule.getEndHour() + ":"
            + selectedSchedule.getEndMinute() + " ");
    }
}

```

```

        + selectedSchedule.title, x + 60, y);
    } else {
        graphics.drawString(editData.getStartHour() + ":"
            + editData.getStartMinute() + " - ", x + 60, y);
        if (state == STATE_SELECT_END) {
            graphics.drawString(editData.getEndHour() + ":"
                + editData.getEndMinute(), x + 108, y);
        }
    }
}

x = 5;
y += fontHeight + 2;

int numSchedule = 0;
int column = 1;
for (int i = 0; i < schedule.length; i++) {
    if (schedule[i].date.compareTo(currentDay) == 0) {
        if (cursorState == ON_SCHEDULE
            && selectedSchedule == schedule[i]) {
            graphics.setColor(Graphics.getColorOfName(Graphics.RED));
        } else {
            graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
        }
        graphics.drawString(schedule[i].getScheduleString(), x, y
            + fontHeight * numSchedule);
        numSchedule++;

        // 一日のデータが 8 を越えたら 2 列に表示する(17 以上は表示できない)
        if (numSchedule >= 8 && column == 1) {
            drawVerticalDottedLine(D_WIDTH / 2 - 2, D_HEIGHT / 2 + 20,
                D_WIDTH / 2 - 2, D_HEIGHT);
            x = D_WIDTH / 2 + 5;
            numSchedule = 0;
            column = 2;
        }
    }
}
}

/**
 * 週間スケジュールを描く
 *
 * @param x
 * @param y
 */
public void drawTable(int x, int y) {
    // スケジュール表の枠を描く
    drawTableLine(x, y);
    // スケジュール表の日付を描く
    drawTableDate(x, y);
    // スケジュール表に予約の時間を棒線として描く
    drawTableGraph(x, y);

    // スケジュール表に選択を示す枠やアンカーを描く

```

```

int dayH = fontHeight + 1;
x += 38;
y += fontHeight;
int currentH = dayH * currentDay.day;

if (state == STATE_CALENDAR) {
    // 選択された日に枠を描く
    graphics.setColor(Graphics.getColorOfRGB(0, 200, 0));
    graphics.drawRect(x, y + currentH, 196, dayH);
    graphics.drawRect(x - 1, y + currentH - 1, 196 + 2, dayH + 2);

} else if (state >= STATE_SELECT_START && cursorState != ON_SCHEDULE) {
    // 【 を描く
    int startW = position(editData.startTime);
    graphics.setColor(Graphics.getColorOfRGB(0, 230, 0));
    graphics.fillRect(x + startW - 1, y + currentH - 2, 3, 14);
    graphics.fillRect(x + startW - 2, y + currentH - 2, 10, 4);
    graphics.fillRect(x + startW - 2, y + currentH + 11, 10, 4);

    if (state >= STATE_SELECT_END) {
        // 】 を描く
        int endW = position(editData.endTime);
        graphics.setColor(Graphics.getColorOfRGB(0, 230, 0));
        graphics.fillRect(x + endW - 1, y + currentH - 2, 3, 14);
        graphics.fillRect(x + endW - 7, y + currentH - 2, 10, 4);
        graphics.fillRect(x + endW - 7, y + currentH + 11, 10, 4);

        // 【 】 の間に棒線を書く
        graphics.setColor(Graphics.getColorOfRGB(190, 190, 190));
        int timeW = position(editData.endTime)
            - position(editData.startTime);
        graphics.fillRect(x + startW, y + currentH + 4, timeW, 6);
    }
}

}

/**
 * 表においての時間の位置を計算する
 */
public int position(int time) {
    int x, distance;
    int halfWidth = (fontHeight + 2) / 2;

    x = time - 900;
    distance = (x / 100 * 2);
    if (x % 100 != 0) {
        distance++;
    }

    return distance * halfWidth;
}

/**
 * スケジュール表を描く 横：1時間 = 14ドット 縦：一日 = 14ドット(fontHeight+2)

```

```

*/
public void drawTableLine(int baseX, int baseY) {
    graphics.setColor(FONT_COLOR);
    graphics.drawString("9", baseX + 36, baseY + 9);
    graphics.drawString("12", baseX + 76, baseY + 9);
    graphics.drawString("15", baseX + 118, baseY + 9);
    graphics.drawString("18", baseX + 160, baseY + 9);
    graphics.drawString("21", baseX + 202, baseY + 9);
    graphics.drawString("23", baseX + 228, baseY + 9);

    int originX = baseX + 38;
    int originY = baseY + fontHeight;
    int hourW = 14;
    int dayH = fontHeight + 1;

    // 横線を引く
    for (int i = 0; i < 8; i++) {
        int height = i * dayH;
        graphics.drawLine(originX, originY + height, originX + 196, originY
            + height);
    }

    // 1時間区切りの線を描く
    for (int i = 0; i < 15; i++) {
        int width = i * hourW;
        graphics.drawLine(originX + width, originY - 2, originX + width,
            originY + dayH * 7);
    }

    // 30分区切りの線を描く
    for (int i = 0; i < 15; i++) {
        int width = i * hourW;
        drawVerticalDottedLine(originX + width + hourW / 2, originY,
            originX + width + hourW / 2, originY + dayH * 7);
    }
}

/**
 * 日にちと曜日を描く
 */
public void drawTableDate(int x, int y) {
    // 書き始める日付を計算
    DayData dat = new DayData(currentDay);
    // 日曜に戻る
    dat.move(-dat.day);

    y += fontHeight;
    // 日曜を描く
    graphics.setColor(Graphics.getColorOfName(Graphics.RED));
    graphics.drawString(dat.getDateString(), x, y + fontHeight);
    dat.move(1);
    y += fontHeight + 1;

    // 平日を描く

```

```

graphics.setColor(Graphics.getColorOfName(Graphics.BLACK));
for (int i = 0; i < 5; i++) {
    graphics.drawString(dat.getDateString(), x, y + fontHeight);
    dat.move(1);
    y += fontHeight + 1;
}

// 土曜を描く
graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
graphics.drawString(dat.getDateString(), x, y + fontHeight);
}

/**
 * 棒線を描く
 */
private void drawTableGraph(int x, int y) {
    int originX = x + 38;
    int originY = y + fontHeight;

    // スケジュールの時間を表す棒を描く
    graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));

    // 書き始める日付、書き終わる日付を計算
    DayData startDate = new DayData(currentDay);
    DayData endDate = new DayData(currentDay);
    startDate.move(-startDate.day);
    endDate.move(6 - endDate.day);

    for (int i = 0; i < schedule.length; i++) {
        if (schedule[i].date.compareTo(startDate) >= 0
            && schedule[i].date.compareTo(endDate) <= 0) {
            int startW = position(schedule[i].startTime);
            int timeW = position(schedule[i].endTime)
                - position(schedule[i].startTime);

            graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
            graphics.fillRect(originX + startW, originY + 4 + 13
                * schedule[i].date.day, timeW, 6);

            // 選択中の予定に枠を描く
            if (state >= STATE_SELECT_START
                && currentDay.compareTo(schedule[i].date) == 0
                && editData.startTime >= schedule[i].startTime
                && editData.startTime < schedule[i].endTime) {
                graphics.setColor(Graphics.getColorOfRGB(0, 230, 0));
                graphics.drawRect(originX + startW, originY + 13
                    * schedule[i].date.day, timeW, 14);
                graphics.drawRect(originX + startW - 1, originY - 1 + 13
                    * schedule[i].date.day, timeW, 14);
                selectedSchedule = schedule[i];
                cursorState = ON_SCHEDULE;
            }
        }
    }
}

```



```

}

/**
 * スケジュール編集画面を描く
 */
public void drawEdit() {
    graphics.setColor(BG_COLOR);
    graphics.fillRect(0, 0, D_WIDTH, D_HEIGHT);

    graphics.setColor(FONT_COLOR);
    graphics.drawString(currentDay.getMonthString(), 5, 5 + fontHeight);

    graphics.setColor(Graphics.getColorOfName(Graphics.BLUE));
    graphics.drawString("名前を入力してください", 76, 5 + fontHeight);

    int x = 1;
    int y = 20;

    // スケジュール表の枠を描く
    drawTable(x, y);

    x += 9;
    y += 125;
    graphics.setColor(FONT_COLOR);
    // 日付を描画する
    graphics.drawString(currentDay.getMonthDateString(), x, y);
    graphics.drawString(editData.getStartHour() + ":"
        + editData.getStartMinute() + " - " + editData.getEndHour()
        + ":" + editData.getEndMinute(), x + 70, y);

    // 予定の名前の入力フォームを描画する
    int c = 12;
    y += 3;
    graphics.setColor(FONT_COLOR);
    graphics.drawString("名前", x, y + fontHeight + 3);
    x += fontHeight * 2 + 3;
    drawBox(editData.title, x, y, c, 1, editCursor == 0);

    y += fontHeight + 8;
    x = 10;
    if (mode == MODE_INSERT) {
        drawButton("登録", x, y, editCursor == 1);
        return;
    } else {
        String str = "変更";
        drawButton(str, x, y, editCursor == 1);
        x += (str.length() + 1) * fontHeight;
        drawButton("削除", x, y, editCursor == 2);
    }
}

/**
 * テキストフォームを描く
 */

```

```

void drawBox(String str, int x, int y, int col, int row, boolean focus) {
    int w = col * fontHeight / 2 + 6;
    int h = row * (fontHeight + 1) + 3;

    if (focus) {
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    } else {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }
    graphics.drawLine(x, y + 1, x, y + h - 1);
    graphics.drawLine(x + w, y + 1, x + w, y + h - 1);
    graphics.drawLine(x + 1, y, x + w - 1, y);
    graphics.drawLine(x + 1, y + h, x + w - 1, y + h);

    graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));

    int sy = y + fontHeight + 3;
    int idx = 0;
    while (sy < y + h && idx < str.length()) {
        int idx2 = font.getLineBreak(str, 0, str.length(), w - 6);
        int idx3 = str.indexOf('\n', idx);
        if (idx3 >= 0 && idx3 < idx2) {
            idx2 = idx3 + 1;
        }
        graphics.drawString(str.substring(idx, idx2), x + 3, sy);
        sy += fontHeight + 1;
        idx = idx2;
    }
}

/**
 * ボタンを描く
 */
void drawButton(String str, int x, int y, boolean focus) {
    if (focus) {
        graphics.setColor(Graphics.getColorOfRGB(255, 0, 0));
    } else {
        graphics.setColor(Graphics.getColorOfRGB(0, 0, 0));
    }

    int width = str.length() * fontHeight;
    graphics.drawRect(x, y, width + 4, fontHeight + 4);
    graphics.drawString(str, x + 3, y + 2 + fontHeight);
}

/**
 * 縦の点線を描く
 *
 * @param x1
 * @param y1
 * @param x2
 * @param y2
 */
public void drawVerticalDottedLine(int x1, int y1, int x2, int y2) {

```

```

        graphics.setColor(Graphics.getColorOfName(Graphics.GRAY));
        for (int i = 0; i < y2 - y1; i += 4) {
            int j = i + 2;
            graphics.drawLine(x1, y1 + i, x2, y1 + j);
        }
    }

    /**
     * スクラッチパッド
     */
    private static final int SP_SCHEDULE_DATA = 0;

    private static final int SP_SCHEDULE_DATA_LENGTH = 102400;

    ScheduleData[] schedule;

    DayData[] allDaySchedules;

    /**
     * データを読み込む
     */
    void loadSchedule() {
        int counter = 0;
        try {
            ByteArrayInputStream is = new ByteArrayInputStream(load(
                SP_SCHEDULE_DATA, SP_SCHEDULE_DATA_LENGTH));
            int num = (is.read() << 8) | is.read();
            schedule = new ScheduleData[num];
            for (int i = 0; i < schedule.length; i++) {
                schedule[i] = ScheduleData.read(is);
                if (schedule[i].isAlldaySchedule()) {
                    counter++;
                }
            }
            is.close();
        } catch (Throwable e) {
            System.out.println("" + e);
        }

        // 一日中予定が入っている日を抽出する
        allDaySchedules = new DayData[counter];

        int k = 0;
        for (int i = 0; i < schedule.length; i++) {
            if (schedule[i].isAlldaySchedule()) {
                allDaySchedules[k] = schedule[i].date;
                k++;
            }
        }
    }
}

```

```

/**
 * データを保存する
 *
 * @param string
 */
public boolean saveScheduleData(ScheduleData data) {
    deleteScheduleData(data);

    // 1 ヶ月以上前のデータを消す
    int counter = 0;
    for (int i = 0; i < schedule.length; i++) {
        if (today.month == 1 && today.compareTo(schedule[i].date) > 8900) {
            counter++;
        } else if (today.compareTo(schedule[i].date) > 100) {
            counter++;
        } else {
            break;
        }
    }
    ScheduleData[] temp = schedule;
    schedule = new ScheduleData[temp.length - counter];
    for (int i = 0; i < schedule.length; i++) {
        schedule[i] = temp[counter + i];
    }

    if (mode != MODE_DELETE) {
        ScheduleData[] dummy = schedule;
        schedule = new ScheduleData[dummy.length + 1];
        int n = 0;
        boolean added = false;
        for (int i = 0; i < dummy.length; i++) {
            if (!added) {
                int def = dummy[i].date.compareTo(data.date);
                if (def > 0 || def == 0
                    && dummy[i].startTime > data.startTime) {
                    schedule[n++] = data;
                    added = true;
                }
            }
            schedule[n++] = dummy[i];
        }
        if (!added) { // どのデータよりも時間が大きい場合はここ
            schedule[n++] = data;
        }
    }

    return saveToScratchpad();
}

// スケジュールデータの削除
void deleteScheduleData(ScheduleData data) {
    int idx = -1;
    for (int i = 0; i < schedule.length; i++) {
        if (schedule[i] == data) {

```

```

        idx = i;
        break;
    }
}
if (idx < 0) {
    return;
}

ScheduleData[] dummy = schedule;
schedule = new ScheduleData[dummy.length - 1];
int n = 0;
for (int i = 0; i < dummy.length; i++) {
    if (i != idx) {
        schedule[n++] = dummy[i];
    }
}
}

boolean saveToScratchpad() {
    try {
        boolean isSavable = false;
        ByteArrayOutputStream os = new ByteArrayOutputStream();
        while (isSavable == false) {
            os.reset();
            os.write(schedule.length >> 8);
            os.write(schedule.length);
            for (int i = 0; i < schedule.length; i++) {
                schedule[i].write(os);
            }

            if (os.toByteArray().length > SP_SCHEDULE_DATA_LENGTH) {
                // データがスクラッチパッドの容量を超えたら古いデータを消す
                deleteScheduleData(schedule[0]);
            } else {
                isSavable = true;
            }
        }

        byte[] b = os.toByteArray();
        os.close();

        if (b.length > SP_SCHEDULE_DATA_LENGTH) {
            return false;
        }

        save(b, SP_SCHEDULE_DATA);
    } catch (Throwable e) {
    }
    return false;
}

// スクラッチパッドへの書き込み
void save(byte[] dat, int pos) {
    OutputStream os = null;

```

```

    try {
        os = Connector.openOutputStream("scratchpad:///0;pos=" + pos);
        os.write(dat);
    } catch (Throwable e) {
    }
    try {
        os.close();
    } catch (Throwable e) {
    }
}

// スクラッチパッドからの読み込み
byte[] load(int pos, int len) {
    InputStream is = null;
    byte[] ret = new byte[len];
    try {
        is = Connector.openInputStream("scratchpad:///0;pos=" + pos);
        is.read(ret);
    } catch (Throwable e) {
        ret = null;
    }
    try {
        is.close();
    } catch (Throwable e) {
    }
    return ret;
}

/** ***** */

// イベント*****
int keyParam = -1;

public void processEvent(int type, int param) {
    if (type == Display.KEY_PRESSED_EVENT) {
        keyParam = param;
    }
}

// キー状態の取得
int getKeyState() {
    int ret = keyParam;
    keyParam = -1;
    return ret;
}

// キー状態のクリア
void clearKeyState() {
    keyParam = -1;
}

// その他処理*****
void sleep(int time) {
    try {

```

```

        Thread.sleep(time);
    } catch (Throwable e) {
    }
}

boolean waitIME = false;

String imeString = null;

/**
 * 入力された文字を取得する
 *
 * @param str
 * @param mode
 * @return
 */
String inputString(String str, int mode) {
    while (getKeypadState() > 0) {
        sleep(100);
    }

    waitIME = true;
    imeOn(str, TextBox.DISPLAY_ANY, mode);

    while (waitIME) {
        sleep(100);
    }

    while (getKeypadState() > 0) {
        sleep(100);
    }
    return imeString;
}

public void processIMEEvent(int type, String str) {
    imeString = str;
    waitIME = false;
}
}

```

[DayData.java]

```

import java.util.Calendar;
import java.util.Date;

public class DayData {
    int year; // 年
    int month; // 月 : 1月が 1
    int date; // 日
    int day; // 曜日 : 日曜が 0

    DayData() {
        this(System.currentTimeMillis());
    }
}

```

```

}

DayData(long time) {
    Calendar cal = Calendar.getInstance();
    cal.setTime(new Date(time));
    year = cal.get(Calendar.YEAR);
    month = cal.get(Calendar.MONTH) + 1;
    date = cal.get(Calendar.DAY_OF_MONTH);
    day = cal.get(Calendar.DAY_OF_WEEK) - Calendar.SUNDAY;
}

DayData(DayData d) {
    this.year = d.year;
    this.month = d.month;
    this.date = d.date;
    this.day = d.day;
}

// 日付を引数だけ進める。引数がマイナスの場合戻る
void move(int def) {
    while (def > 0) {
        date++;
        if (date > getDaysInMonth(year, month)) {
            date = 1;
            month++;
            if (month > 12) {
                month = 1;
                year++;
            }
        }
        day = (day + 1) % 7;
        def--;
    }
    while (def < 0) {
        date--;
        if (date < 1) {
            month--;
            if (month < 1) {
                month = 12;
                year--;
            }
            date = getDaysInMonth(year, month);
        }
        day = (day + 6) % 7;
        def++;
    }
}

// 日付の比較。dのほうが大きい場合はマイナス。 this-d と覚えれば OK
int compareTo(DayData d) {
    int prm1 = this.year * 10000 + this.month * 100 + this.date;
    int prm2 = d.year * 10000 + d.month * 100 + d.date;
    return prm1 - prm2;
}

```



```

// 年と月から、その月の日数を計算する
static final int[] DAYS_IN_MONTH = { // それぞれの月の日数
31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, };

static int getDaysInMonth(int y, int m) {
    if (m == 2 && y % 4 == 0) { // 100年に一度とか、2000年に一度は無視
        return 29;
    }
    return DAYS_IN_MONTH[m - 1];
}

// 文字列に変換
String getMonthString() {
    return "" + year + "年" + (month / 10) + (month % 10) + "月";
}

// 月と曜日を文字列に変換する
String getMonthDateString() {
    return month + "月" + date + "(" + getKanjiOf(day) + ")";
}

// 日と曜日を文字列に変換する
String getDateString() {
    String preposition = "";
    if(date < 10){
        preposition = " ";
    }
    return preposition + date + "(" + getKanjiOf(day) + ")";
}

// 曜日を漢字に変換する
String getKanjiOf(int day){
    switch (day){
        case 0:
            return "日";
        case 1:
            return "月";
        case 2:
            return "火";
        case 3:
            return "水";
        case 4:
            return "木";
        case 5:
            return "金";
        case 6:
            return "土";
        default:
            return "";
    }
}
}

```

```

    public String getYear() {
        return Integer.toString(year);
    }

    public String getDate() {
        return "" + (date / 10) + (date % 10);
    }

    public String getMonth() {
        return "" + (month / 10) + (month % 10);
    }
}

```

[ScheduleData.java]

```

import java.io.InputStream;
import java.io.OutputStream;

public class ScheduleData {
    int id;
    DayData date;           //日付
    int startTime, endTime; //開始時刻、終了時刻。HHMM と入る
    String title;          //タイトル
    boolean profile;       //予定の内容

    //定数
    static final boolean RESERVE = true; //予約
    static final boolean PLAN = false;  //用事

    //コンストラクタ
    ScheduleData() {
        date = new DayData();
        startTime = 900;
        endTime = 1000;
        title = "";
        profile = RESERVE;
    }

    //データの書き込み
    void write(OutputStream os) throws Exception {
        os.write(date.year >> 8);
        os.write(date.year);
        os.write(date.month);
        os.write(date.date);
        os.write(date.day);

        os.write(startTime >> 8);
        os.write(startTime);
        os.write(endTime >> 8);
        os.write(endTime);

        if(profile){

```

```

        os.write(1);
    }else{
        os.write(0);
    }

        os.write(id >> 24);
        os.write(id >> 16);
        os.write(id >> 8);
        os.write(id);

        byte[] b = title.getBytes();
        os.write(b.length >> 8);
        os.write(b.length);
        os.write(b);
    }

//データの読み込み
static ScheduleData read(InputStream is) throws Exception {
    ScheduleData sd = new ScheduleData();
    sd.date.year = (is.read() << 8) | is.read();
    sd.date.month = is.read();
    sd.date.date = is.read();
    sd.date.day = is.read();

    sd.startTime = (is.read() << 8) | is.read();
    sd.endTime = (is.read() << 8) | is.read();

    if(is.read() == 1){
        sd.profile = true;
    }else{
        sd.profile = false;
    }

    sd.id = (is.read() << 24) | (is.read() << 16) | (is.read() << 8) | is.read();

    int len = (is.read() << 8) | is.read();
    byte[] b = new byte[len];
    is.read(b);
    sd.title = new String(b);

    return sd;
}

public String getScheduleString(){
    return getStartHour() + ":" + getStartMinute() + " " + title;
}

//開始時間を変更する
public void shiftStartTime(int i){
    if(i > 0){ //増やす
        if(startTime >= 2230){
            startTime = 900;
        }else if(startTime % 100 > 0){ //30分だったら

```

```

        //xx:00 分にする
        startTime = (startTime / 100) * 100;
        //1 時間足す
        startTime += 100;
    }else if(startTime % 100 == 0){ //0 分だったら
        //30 分足す
        startTime += 30;
    }
}else if(i < 0){ //減らす
    if(startTime == 900){
        startTime = 2230;
    }else if(startTime % 100 > 0){ //30 分だったら
        //xx:00 分にする
        startTime = (startTime / 100) * 100;
    }else if(startTime % 100 == 0){ //0 分だったら
        //30 分足す
        startTime += 30;
        //1 時間減らす
        startTime -= 100;
    }
}
}

//終了時間を変更する
public void shiftEndTime(int i){
    if(i > 0){ //増やす
        if(endTime < 2300){
            if(endTime % 100 > 0){ //30 分だったら
                //xx:00 分にする
                endTime = (endTime / 100) * 100;
                //1 時間足す
                endTime += 100;
            }else if(endTime % 100 == 0){ //0 分だったら
                //30 分足す
                endTime += 30;
            }
        }
    }else if(i < 0){ //減らす
        if(endTime > startTime + 70){
            if(endTime % 100 > 0){ //30 分だったら
                //xx:00 分にする
                endTime = (endTime / 100) * 100;
            }else if(endTime % 100 == 0){ //0 分だったら
                //30 分足す
                endTime += 30;
                //1 時間減らす
                endTime -= 100;
            }
        }
    }
}

String getStartHour() {
    int h = startTime / 100;

```

```

        return "" + (h / 10) + (h % 10);
    }

    String getStartMinute() {
        int m = startTime % 100;
        return "" + (m / 10) + (m % 10);
    }

    String getEndHour() {
        int h = endTime / 100;
        return "" + (h / 10) + (h % 10);
    }

    String getEndMinute() {
        int m = endTime % 100;
        return "" + (m / 10) + (m % 10);
    }

    public void setEndTime(int endTime) {
        this.endTime = endTime;
    }

    public void setStartTime(int startTime) {
        if(startTime > 2230){
            this.startTime = 900;
        }else{
            this.startTime = startTime;
        }
    }

    boolean isAlldaySchedule(){
        if(startTime == 900 && endTime == 2300){
            return true;
        }else{
            return false;
        }
    }
}

```

1.3. UC05 サイト構築

[index.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <META http-equiv="Content-Style-Type" content="text/css">
  <title>たかくさき療術院 -トップページ</title>
  <META name="Author" content="たかくさき療術院">
  <META name="Description" content="たかくさき療術院のホームページです。こち
らは療術院紹介のページです。当院では機械を使用せずに、夫婦の手だけでマッサージをするの
で、子供から高齢者まで、また妊娠の方などでも誰でもマッサージを受けることができます。日々
蓄積される体の凝りを解消し、体本来の治癒力を高め、頭痛、腰痛、肩こりから抜け毛、生理不
順、不妊などの症状を改善しましょう。">
  <META name="keywords" content="たかくさき, たかくさき療術院,療術院,マッサー
ジ,神奈川県,横浜,本郷台, 自然治癒力, 体のメンテナンス, 4つの手, 抜け毛, 頭痛, 肩こり, 腰痛, 不
妊, 出張">
  <Meta name="robots" content="index,nofollow">
<style type="text/css">
<!--
#ImgContainer {height:110px;}
img{border:none;}
font{border:none;}
-->
</style>

<script type="text/javascript">
<!--

var img1_on = new Image();
var img1_off = new Image();
var img2_on = new Image();
var img2_off = new Image();
var img3_on = new Image();
var img3_off = new Image();
var img4_on = new Image();
var img4_off = new Image();
var img5_on = new Image();
var img5_off = new Image();
img1_on.src="images/top-button02.gif";
img1_off.src="images/top-button03.gif";
img2_on.src="images/intro-button02.gif";
img2_off.src="images/intro-button03.gif";
img3_on.src="images/access-button02.gif";
img3_off.src="images/access-button03.gif";
img4_on.src="images/product-button02.gif";
img4_off.src="images/product-button03.gif";
img5_on.src="images/reserve-button02.gif";
img5_off.src="images/reserve-button03.gif";

function rollon(targetimage){

  if(targetimage == "image1"){
```

```

        document.images[targetimage].src=img1_on.src;

    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_on.src;

    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_on.src;

    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_on.src;

    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_on.src;

    }
}

function rolloff(targetimage){

    if(targetimage == "image1"){
        document.images[targetimage].src=img1_off.src;

    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_off.src;

    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_off.src;

    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_off.src;

    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_off.src;

    }

}

</script>
</head>

<body bgcolor="#CC6633" background="images/back-image.gif" text="#CC6633"
link="#CC6633" vlink="#CC6633" alink="#CC6633" topmargin="0" leftmargin="0"
marginwidth="0" marginheight="0">
<SCRIPT LANGUAGE="JAVASCRIPT"
SRC="http://j1.ax.xrea.com/l.j?id=100412602"></SCRIPT>
<NOSCRIPT><A HREF="http://w1.ax.xrea.com/c.f?id=100412602"
TARGET="_blank"><IMG SRC="http://w1.ax.xrea.com/l.f?id=100412602&url=X"
BORDER="0"></A></NOSCRIPT>
<!-- バナー -->
<table width="800" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="800" height="100" colspan="5"><IMG
src="images/banner02.gif"> </td>

```

```

        </tr>
</table>

<!-- 上部のメニューバー -->
<table align="center" width="800" height="40" border="0" cellspacing="0"
cellpadding="0" background="images/opt-bar02.gif">
    <tr>
        <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
        <td width="131" height="40" align="center"><a
href="index.html"><IMG name="image1" src="images/top-button03.gif"
onMouseOut="rolloff('image1')" onMouseOver="rollon('image1')"></a></td>
        <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
        <td width="131" height="40" align="center"><a
href=intro.html><IMG name="image2" src="images/intro-button03.gif" alt="療術院紹介"
onMouseOut="rolloff('image2')" onMouseOver="rollon('image2')"></a></td>
        <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
        <td width="121" height="40" align="center"><a
href=access.html><IMG name="image3" src="images/access-button03.gif" alt="来院方法"
onMouseOut="rolloff('image3')" onMouseOver="rollon('image3')"></a></td>
        <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
        <td width="121" height="40" align="center"><a
href=products.html><IMG name="image4" src="images/product-button03.gif" alt="商品紹介"
onMouseOut="rolloff('image4')" onMouseOver="rollon('image4')"></a></td>
        <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
        <td width="170" height="40" align="center"><a
href=reserve.html><IMG name="image5" src="images/reserve-button03.gif" alt="予約・お問
い合わせ" width="170" onMouseOut="rolloff('image5')"
onMouseOver="rollon('image5')"></a></td>
        <td width="21" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    </tr>
</table>

<!-- 紹介文 -->
<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td colspan="2"><IMG src="images/spacer.gif"></td>
    </tr>
    <tr>
        <td><IMG src="images/takakusaki-pic.jpg" width="280"></td>
        <td>
            <table align="center" width="500" height="200"
border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td><h3>わたしたちは、機械などは使用せずに、
二人合わせて4つの手で全身のマッサージを行います。体の凝りによって現れる症状は様々です。
<br>
                    頭痛、こり、抜け毛や生理不順まで。<br>
                    体の凝りをほぐし、体本来の治癒力を発揮させま
                </td>
            </table>
        </td>
    </tr>
</table>

```


しょう！

機械を使わない施術方法なので子供から高齢者まで、また妊娠中の方等でも誰でも安心して受けられます。</h3></td>

</tr>

</table>

</td>

</tr>

</table>

<!-- 住所・時間等 -->

<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">

<tr>

<td colspan="3"></td>

</tr>

<tr>

<td width="200" height="200">

<table width="200" height="200" align="center"

border="0" cellpadding="0" cellspacing="0">

<tr>

<td width="200" height="100"><IMG

src="images/valentine.jpg" width="150" height="100"></td>

</tr>

<tr>

<td width="150" height="50"

align="left">体の痛みを解消して、大切な人との時間を思いっきり満喫

</td>

</tr>

</table>

</td>

<td width="500" height="200">

<table width="500" height="200" align="center"

valign="center" border="0" cellspacing="0" cellpadding="0">

<tr>

<td width="300" align="left"><IMG

src="images/open-hours.gif"></td>

<td width="200" align="left"><IMG

src="images/closed-days2.gif"></td>

</tr>

<tr>

<td colspan="2" width="500"

align="left"></td>

</tr>

</table>

</td>

<td width="100" align="center" valign="center"></td>

</tr>

<!-- 下部メニューバー -->

</table>

<table width="800" align="center">

<tr>

<td width="800"></td>

</tr>

<tr>

```

        <td align="center">|<a href="index.html"> トップページ
</a>|<a href="intro.html"> 療術院紹介 </a>|
        <a href="access.html"> 来院方法</a>|<a href="products.html">
商品紹介 </a>|<a href="reserve.html"> ご予約・お問い合わせ </a>|</td>
        </tr>
</table>
</body>
</html>

```

[intro.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
    <head>
        <META http-equiv="Content-Style-Type" content="text/css">
        <title>たかくさき療術院 -療術院紹介</title>
        <META name="Author" content="たかくさき療術院">
        <META name="Description" content="たかくさき療術院のホームページで
す。こちらはトップページです。当院では機械を使用せずに、夫婦の手だけでマッサージをする
ので、子供から高齢者まで、また妊娠の方などでも誰でもマッサージを受けることができます。
日々蓄積される体の凝りを解消し、体本来の治癒力を高め、頭痛、腰痛、肩こりから抜け毛、生
理不順、不妊などの症状を改善しましょう。">
        <META name="keywords" content="たかくさき, たかくさき療術院,療術院,
マッサージ,神奈川県,横浜,本郷台, 自然治癒力, 体のメンテナンス, 4つの手,抜け毛,頭痛,肩こり,
腰痛,不妊,出張">
        <META name="robots" content="index,nofollow">
    <style type="text/css">
    <!--
#ImgContainer {height:110px;}
img{border:none;}
-->
</style>

    <script type="text/javascript">
    <!--

var img1_on = new Image();
var img1_off = new Image();
var img2_on = new Image();
var img2_off = new Image();
var img3_on = new Image();
var img3_off = new Image();
var img4_on = new Image();
var img4_off = new Image();
var img5_on = new Image();
var img5_off = new Image();
img1_on.src="images/top-button02.gif";
img1_off.src="images/top-button03.gif";
img2_on.src="images/intro-button02.gif";
img2_off.src="images/intro-button03.gif";
img3_on.src="images/access-button02.gif";
img3_off.src="images/access-button03.gif";

```

```

img4_on.src="images/product-button02.gif";
img4_off.src="images/product-button03.gif";
img5_on.src="images/reserve-button02.gif";
img5_off.src="images/reserve-button03.gif";

function rollon(targetimage){
    if(targetimage == "image1"){
        document.images[targetimage].src=img1_on.src;
    }
    else if(targetimage == "image2"){
        document.images[targetimage].src=img2_on.src;
    }
    else if(targetimage == "image3"){
        document.images[targetimage].src=img3_on.src;
    }
    else if(targetimage == "image4"){
        document.images[targetimage].src=img4_on.src;
    }
    else if(targetimage == "image5"){
        document.images[targetimage].src=img5_on.src;
    }
}

function rolloff(targetimage){
    if(targetimage == "image1"){
        document.images[targetimage].src=img1_off.src;
    }
    else if(targetimage == "image2"){
        document.images[targetimage].src=img2_off.src;
    }
    else if(targetimage == "image3"){
        document.images[targetimage].src=img3_off.src;
    }
    else if(targetimage == "image4"){
        document.images[targetimage].src=img4_off.src;
    }
    else if(targetimage == "image5"){
        document.images[targetimage].src=img5_off.src;
    }
}

</script>
</head>
<body bgcolor="#FFFF99" background="images/back-image.gif" text="#000000"
link="#CC6633" vlink="#CC6633" alink="#CC6633" topmargin="0" leftmargin="0"
marginwidth="0" marginheight="0">
    <!-- バナー -->
    <table width="800" align="center" cellpadding="0" cellspacing="0">
        <tr>

```

```

                <td width="800" height="100" colspan="5"><a
name="pagetop"><IMG src="images/banner02.gif"></a></td>
            </tr>
        </table>

        <!-- 上部メニューバー -->
        <table align="center" width="800" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td height="40" align="center"
background="images/opt-bar02.gif">
                    <table align="center" border="0" cellspacing="0"
cellpadding="0">
                        <tr>
                            <td width="15" height="40"
align="center"><IMG src="images/bar-spacer03.gif" height="25"></td>
                            <td width="131" height="40"
align="center"><a href="index.html"><IMG name="image1" src="images/top-button03.gif"
onMouseOut="rolloff('image1')" onMouseOver="rollon('image1')"></a></td>
                            <td width="15" height="40"
align="center"><IMG src="images/bar-spacer03.gif" height="25"></td>
                            <td width="131" height="40"
align="center"><a href=intro.html><IMG name="image2" src="images/intro-button03.gif"
alt="療術院紹介" onMouseOut="rolloff('image2')"
onMouseOver="rollon('image2')"></a></td>
                            <td width="15" height="40"
align="center"><IMG src="images/bar-spacer03.gif" height="25"></td>
                            <td width="121" height="40"
align="center"><a href=access.html><IMG name="image3"
src="images/access-button03.gif" alt="来院方法" onMouseOut="rolloff('image3')"
onMouseOver="rollon('image3')"></a></td>
                            <td width="15" height="40"
align="center"><IMG src="images/bar-spacer03.gif" height="25"></td>
                            <td width="121" height="40"
align="center"><a href=products.html><IMG name="image4"
src="images/product-button03.gif" alt="商品紹介" width="121"
onMouseOut="rolloff('image4')" onMouseOver="rollon('image4')"></a></td>
                            <td width="15" height="40"
align="center"><IMG src="images/bar-spacer03.gif" height="25"></td>
                            <td width="170" height="40"
align="center"><a href=reserve.html><IMG name="image5"
src="images/reserve-button03.gif" alt="予約・お問い合わせ" width="170"
onMouseOut="rolloff('image5')" onMouseOver="rollon('image5')"></a></td>
                            <td width="21" height="40"
align="center"><IMG src="images/bar-spacer03.gif" height="25"></td>
                        </tr>
                    </table>
                </td>
            </tr>
        </table>

        <!-- 本文 -->
        <table align="center" width="800" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td width="800" colspan="2"><IMG
src="images/spacer.gif"></td>
            </tr>
        </table>

```

```

</tr>
<tr>
  <td valign="top" align="left" width="200" border="0"
cellspacing="0" cellpadding="0"><a href="#sejyutu"><IMG
src="images/sejyutu-icon01.gif"></a><br>
  <a href="#for"><IMG
src="images/for-icon01.gif"></a><br>
  <a href="#charges"><IMG
src="images/charges-icon01.gif"></a>
</td>
<td>
  <table width="600" height="250" border="0"
align="center" cellspacing="0" cellpadding="0">
    <tr>
      <td align="left"><a
name="sejyutu"><IMG src="images/title-sideicon.gif"></a><font size="6">施術方法
</font></td>
    </tr>
    <tr>
      <td align="center"><IMG
src="images/dottedline.gif" width="600"></td>
    </tr>
    <tr>
      <td>
        <table width="580" height="200"
align="center" border="0">
          <tr>
            <td><font color="#CC6633"
size="4">～ たいま工事中 ～ <br><br>
              私たちは機械を全
              く使用せず、4つの手のみでマッサージを行います。体の凝りを敏感に感じ取り、また解消する
              には、やはり人間の手が一番だと信じているからです。<br><br>
              肩が凝っているか
              ら肩だけをマッサージする！のではありません。<br>体の一部だけをほぐしても、それは一時
              的に楽になるだけです。日々の生活で蓄積された体全体の凝りを解消して、バランスを整えなけ
              れば凝りは再発します。
              だから私達は1時間
              以上かけて、体全体をマッサージします。体全体の筋肉の凝りをほぐし、血行を促進させること
              により、体本来の治癒力を発揮させることができます。<br><br>
              またマッサージだ
              けではありません。私達は患者様に「心身ともに」健康でいて欲しいと願っています。マッサー
              ジの前後に暖かいお茶を飲みながらちょっと雑談や相談。そんなゆったりした時間の中で、普段
              忙しい人もちょっと心にゆとりが。
              だからこそ私達は
              一人あたり1時間半程の時間確保したいと考えています。中には忙しい方もいらっしゃるので、
              もちろん強制ではありません。1時間でも予約を承ります。<br>
              <br>
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</td>

```

```

        </tr>
        <tr>
            <td width="800" colspan="2"><IMG
src="images/spacer.gif"></td>
        </tr>
        <tr>
            <td width="200"></td>
            <td>
                <table width="600" height="250" border="0"
cellspacing="0" cellpadding="0">
                    <tr>
                        <td align="left"><a
name="for"><IMG src="images/title-sideicon.gif"></a><font size="6">こんな症状の方へ
</font></td>
                    </tr>
                    <tr>
                        <td align="center"><IMG
src="images/dottedline.gif" width="600"></td>
                    </tr>
                    <tr>
                        <td>
                            <table width="580" height="200"
align="center" border="0">
                                <tr>
                                    <td><font color="#CC6633"
size="4">～ただいま工事中～<br><br>
                                    肩こりや腰痛は一番分りや
                                    すい体の凝りのサインですね。しかしそれ以外にも、体の凝りが原因のものはたくさんあります。
                                    <br>知ってますか？</font>
                                    <h3>頭痛・目の疲れ・抜け
                                    毛・生理不順・不妊</h3>
                                    <font color="#CC6633"
size="4">こんな症状も、体の筋肉の凝りが原因だということを。<br>
                                    そうです、あきらめていたこ
                                    んな症状も、体のメンテナンスを行うことで治るのです。髪だって生えてくるんです！<br>
                                    他にも様々な症状でお悩みの
                                    方、当院で一度体の凝りをほぐしてみませんか？また症状について、電話でもお気軽にご相談下
                                    さい。</font></td>
                                </tr>
                            </table>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
        <tr>
            <td width="800" colspan="2"><IMG
src="images/spacer.gif"></td>
        </tr>
        <tr>
            <td width="200"></td>
            <td width="600" height="300">
                <table width="600" border="0" cellspacing="0"
cellpadding="0">

```

```

                <tr>
                    <td align="left"><a
name="charges"><IMG src="images/title-sideicon.gif"></a><font size="6">料金案内
</font></td>
                </tr>
                <tr>
                    <td align="center"><IMG
src="images/dottedline.gif" width="600"></td>
                </tr>
                <tr><td width="600"><IMG
src="images/spacer.gif" width="600"></td></tr>
                <tr>
                    <td>
                        <table align="center" width="540"
height="100" border="2" cellspacing="0" cellpadding="0">
                            <tr>
                                <td width="150"
align="center"><font size="6">大人</font></td>
                                <td width="150"
align="center"><font size="6">高校生</font></td>
                                <td width="240"
align="center"><font size="6">中学生・シニア</font></td>
                            </tr>
                            <tr>
                                <td width="150"
align="center"><font size="6"><font size="6">5000 円</font></td>
                                <td width="150"
align="center"><font size="6"><font size="6">4000 円</font></td>
                                <td width="240"
align="center"><font size="6"><font size="6">3000 円</font></td>
                            </tr>
                        </table>
                    </td>
                </tr>
                <tr>
                    <td colspan="2" width="800"><IMG
src="images/spacer.gif"></td>
                </tr>
                <tr>
                    <td width="200"></td>
                    <td align="right"><a href="#pagetop">上へ戻る</a></td>
                </tr>
            </table>

            <!-- 下部メニューバー -->
            <table width="800" align="center">
                <tr>
                    <td align="center">| <a href="index.html"> トップページ
</a> | <a href="intro.html"> 療術院紹介 </a> |
                    <a href="access.html"> 来院方法 </a> | <a
href="products.html"> 商品紹介 </a> | <a href="reserve.html"> ご予約・お問い合わせ

```

```
</a>|</td>
      </tr>
    </table>
</body>
</html>
```

[access.html]

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

```
<html>
  <head>
    <META http-equiv="Content-Style-Type" content="text/css">
    <title>たかくさき療術院 来院方法</title>
    <META name="Author" content="たかくさき療術院">
    <META name="Description" content="たかくさき療術院のホームページで
す。こちらは来院方法のページです。当院では機械を使用せずに、夫婦の手だけでマッサージを
するので、子供から高齢者まで、また妊娠の方などでも誰でもマッサージを受けることができま
す。日々蓄積される体の凝りを解消し、体本来の治癒力を高め、頭痛、腰痛、肩こりから抜け毛、
生理不順、不妊などの症状を改善しましょう。">
    <META name="keywords" content="たかくさき, たかくさき療術院,療術院,
マッサージ,神奈川県,横浜,本郷台, 自然治癒力, 体のメンテナンス, 4つの手,抜け毛,頭痛,肩こり,
腰痛,不妊,出張">
    <META name="robots" content="index,nofollow">
  <style type="text/css">
  <!--
#ImgContainer {height:110px;}
img{border:none;}
-->
</style>
<script type="text/javascript">
  <!--
var img1_on = new Image();
var img1_off = new Image();
var img2_on = new Image();
var img2_off = new Image();
var img3_on = new Image();
var img3_off = new Image();
var img4_on = new Image();
var img4_off = new Image();
var img5_on = new Image();
var img5_off = new Image();
img1_on.src="images/top-button02.gif";
img1_off.src="images/top-button03.gif";
img2_on.src="images/intro-button02.gif";
img2_off.src="images/intro-button03.gif";
img3_on.src="images/access-button02.gif";
img3_off.src="images/access-button03.gif";
img4_on.src="images/product-button02.gif";
img4_off.src="images/product-button03.gif";
img5_on.src="images/reserve-button02.gif";
img5_off.src="images/reserve-button03.gif";
```



```

function rollon(targetimage){
    if(targetimage == "image1"){
        document.images[targetimage].src=img1_on.src;
    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_on.src;
    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_on.src;
    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_on.src;
    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_on.src;
    }
}

function rolloff(targetimage){
    if(targetimage == "image1"){
        document.images[targetimage].src=img1_off.src;
    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_off.src;
    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_off.src;
    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_off.src;
    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_off.src;
    }
}
}

```

```

                <td width="800" height="100" colspan="5"><IMG
src="images/banner02.gif"> </td>
            </tr>
        </table>

        <!-- 上部のメニューバー -->
        <table align="center" width="800" height="40" border="0" cellspacing="0"
cellpadding="0" background="images/opt-bar02.gif">
            <tr>
                <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
                <td width="131" height="40" align="center"><a
href="index.html"><IMG name="image1" src="images/top-button03.gif"
onMouseOut="rolloff('image1')" onMouseOver="rollon('image1')"></a></td>
                <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
                <td width="131" height="40" align="center"><a
href=intro.html><IMG name="image2" src="images/intro-button03.gif" alt="療術院紹介"
onMouseOut="rolloff('image2')" onMouseOver="rollon('image2')"></a></td>
                <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
                <td width="121" height="40" align="center"><a
href=access.html><IMG name="image3" src="images/access-button03.gif" alt="来院方法"
onMouseOut="rolloff('image3')" onMouseOver="rollon('image3')"></a></td>
                <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
                <td width="121" height="40" align="center"><a
href=products.html><IMG name="image4" src="images/product-button03.gif" alt="商品紹介"
onMouseOut="rolloff('image4')" onMouseOver="rollon('image4')"></a></td>
                <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
                <td width="170" height="40" align="center"><a
href=reserve.html><IMG name="image5" src="images/reserve-button03.gif" alt="予約・お問
い合わせ" width="170" onMouseOut="rolloff('image5')"
onMouseOver="rollon('image5')"></a></td>
                <td width="21" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
            </tr>
        </table>

        <!-- 本文 -->
        <table align="center" width="800" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td colspan="2" width="800"><IMG
src="images/spacer.gif"></td>
            </tr>
            <tr>
                <td colspan="2" width="800"><IMG
src="images/spacer.gif"></td>
            </tr>
            <tr>
                <td width="500" height="550" valign="top" align="left" ><IMG
src="images/map.jpg"><br>
                <font size="3" align="left">地図が大変見づらくなっていることに

```

お詫び申し上げます。
ただいま新しい地図との差し替えの途中です。
ご迷惑おかけし
て申し訳ありません。</td>

```
<td width="300" height="550" valign="top" align="center">  
  <table width="300" height="550" align="center"  
    valign="top" border="0" cellspacing="0" cellpadding="0">  
      <tr>  
        <td width="300" height="200">  
          <table align="left"  
            valign="top" width="300" height="150" border="0" cellspacing="0" cellpadding="0">  
              <tr>  
                <td  
                  width="300" valign="top" align="left" cellspacing="0" cellpadding="0"><IMG  
                    src="images/title-sideicon.gif"><font size="6">住所・連絡先</font></td>  
                </tr>  
                <tr>  
                <td  
                  width="300" valign="top" cellspacing="0" cellpadding="0"><IMG src="images/dottedline.gif"  
                    width="300"></td>  
                </tr>  
                <tr>  
                <td  
                  width="300"><IMG src="images/spacer.gif" width="300"></td>  
                </tr>  
                <tr>  
                <td  
                  width="300" height="100" valign="top" align="left" cellspacing="0" cellpadding="0"><font  
                    size="4">〒247-0008</font><br><font size="5">神奈川県横浜市栄区本郷台  
                    3-12-19</font><br>  
                    <IMG  
                      src="images/spacer.gif" width="300"><br><font size="5">tel&fax  
                      045(894)6713</font></td>  
                </tr>  
                <tr>  
                <td><IMG src="images/spacer.gif" width="300"></td>  
                </tr>  
              </table>  
            </td>  
          </tr>  
          <tr>  
            <td width="300" height="50">  
              <table>  
                <tr>  
                <td  
                  width="300" valign="top" align="left" cellspacing="0" cellpadding="0"><IMG  
                    src="images/title-sideicon.gif"><font size="6">車</font></td>  
                </tr>  
                <tr>  
                <td  
                  width="300" valign="top" cellspacing="0" cellpadding="0"><IMG src="images/dottedline.gif"  
                    width="300"></td>  
                </tr>  
                <tr>  
                <td  
                  width="300" height="50" align="center"><IMG src="images/spacer.gif" width="300"></td>  
                </tr>  
              </table>  
            </td>  
          </tr>  
        </table>  
      </tr>  
    </table>
```

```

width="300"><IMG src="images/spacer.gif" width="300"></td>
</tr>
<tr>
<td><font
size="5">駐車 可</font></td>
</tr>
<tr>
<td>
width="300"><IMG src="images/spacer.gif" width="300"></td>
</tr>
</table>
</td>
</tr>
<tr>
<td width="300" height="300"
align="center">
<table align="center"
valign="top" width="300" height="300" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>
width="300"><IMG src="images/spacer.gif" width="300"></td>
</tr>
<tr>
<td>
width="300" height="35" align="left" valign="top" cellspacing="0" cellpadding="0"><IMG
src="images/title-sideicon.gif"><font size="6">最寄り駅</font></td>
</tr>
<tr>
<td>
width="300" valign="top" cellspacing="0" cellpadding="0"><IMG src="images/dottedline.gif"
width="300"></td>
</tr>
<tr>
<td>
width="250" valign="top" cellspacing="0" cellpadding="0"><font size="5">JR 京浜東北線
本郷台駅</font><br><br>
<font size="4">【徒歩】 駅から徒歩 15 分ほど</font><br>
<font size="4">【バス】 本郷台・戸塚・大船駅からのバス便有り<br> 便が複雑です
ので、バスでお越しの方はお問い合わせ下さい。</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

[product.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <META http-equiv="Content-Style-Type" content="text/css">
  <title>たかくさき療術院 -商品紹介</title>
  <META name="Author" content="たかくさき療術院">
  <META name="Description" content="たかくさき療術院のホームページです。こち
らは商品紹介のページです。当院では機械を使用せずに、夫婦の手だけでマッサージをするので、
子供から高齢者まで、また妊娠の方などでも誰でもマッサージを受けることができます。日々蓄
積される体の凝りを解消し、体本来の治癒力を高め、頭痛、腰痛、肩こりから抜け毛、生理不順、
不妊などの症状を改善しましょう。">
  <META name="keywords" content="たかくさき, たかくさき療術院,療術院,マッサー
ジ,神奈川県,横浜,本郷台, 自然治癒力, 体のメンテナンス, 4つの手,抜け毛,頭痛,肩こり,腰痛,不
妊,出張">
  <META name="robots" content="index,nofollow">
<style type="text/css">
<!--
#ImgContainer {height:110px;}
img{border:none;}
-->
</style>
<script type="text/javascript">
<!--
var img1_on = new Image();
var img1_off = new Image();
var img2_on = new Image();
var img2_off = new Image();
var img3_on = new Image();
var img3_off = new Image();
var img4_on = new Image();
var img4_off = new Image();
var img5_on = new Image();
var img5_off = new Image();
img1_on.src="images/top-button02.gif";
img1_off.src="images/top-button03.gif";
img2_on.src="images/intro-button02.gif";
img2_off.src="images/intro-button03.gif";
img3_on.src="images/access-button02.gif";
img3_off.src="images/access-button03.gif";
img4_on.src="images/product-button02.gif";
img4_off.src="images/product-button03.gif";
img5_on.src="images/reserve-button02.gif";
img5_off.src="images/reserve-button03.gif";

function rollon(targetimage){

  if(targetimage == "image1"){
    document.images[targetimage].src=img1_on.src;
```

```

    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_on.src;

    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_on.src;

    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_on.src;

    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_on.src;

    }
}

function rolloff(targetimage){

    if(targetimage == "image1"){
        document.images[targetimage].src=img1_off.src;

    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_off.src;

    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_off.src;

    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_off.src;

    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_off.src;

    }

}

!-->
</script>
</head>
<body bgcolor="#FFFF99" background="images/back-image.gif" text="#000000"
link="#CC6633" vlink="#CC6633" alink="#CC6633" topmargin="0" leftmargin="0"
marginwidth="0" marginheight="0">
<SCRIPT LANGUAGE="JAVASCRIPT"
SRC="http://j1.ax.xrea.com/l.j?id=100412602"></SCRIPT>
<NOSCRIPT><A HREF="http://w1.ax.xrea.com/c.f?id=100412602"
TARGET="_blank"><IMG SRC="http://w1.ax.xrea.com/l.f?id=100412602&url=X"
BORDER="0"></A></NOSCRIPT>
<!-- バナー -->
<table width="800" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td width="800" height="100" colspan="5"><a
name="pagetop"><IMG src="images/banner02.gif"></a></td>
    </tr>
</table>

```

```

<!-- 上部のメニューバー -->
<table align="center" width="800" height="40" border="0" cellspacing="0"
cellpadding="0" background="images/opt-bar02.gif">
  <tr>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="131" height="40" align="center"><a
href="index.html"><IMG name="image1" src="images/top-button03.gif"
onMouseOut="rolloff('image1')" onMouseOver="rollon('image1')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="131" height="40" align="center"><a
href=intro.html><IMG name="image2" src="images/intro-button03.gif" alt="療術院紹介"
onMouseOut="rolloff('image2')" onMouseOver="rollon('image2')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="121" height="40" align="center"><a
href=access.html><IMG name="image3" src="images/access-button03.gif" alt="来院方法"
onMouseOut="rolloff('image3')" onMouseOver="rollon('image3')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="121" height="40" align="center"><a
href=products.html><IMG name="image4" src="images/product-button03.gif" alt="商品紹介"
onMouseOut="rolloff('image4')" onMouseOver="rollon('image4')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="170" height="40" align="center"><a
href=reserve.html><IMG name="image5" src="images/reserve-button03.gif" alt="予約・お問
い合わせ" width="170" onMouseOut="rolloff('image5')"
onMouseOver="rollon('image5')"></a></td>
    <td width="21" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
  </tr>
</table>

<!-- 商品紹介 -->
<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><IMG src="images/spacer.gif"></td>
  </tr>
  <tr>
    <td height="35" align="left" valign="top"><IMG
src="images/title-sideicon.gif"><font size="6">商品紹介</font></td>
  </tr>
  <tr>
    <td align="left" valign="top"><IMG src="images/dottedline.gif"
width="300"></td>
  </tr>
</table>

<table width="800" align="center" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td align="left" width="200" height="180" align="center"><IMG
src="images/socks.jpg"></td>

```

<p>る“5本指靴下”
</p>	<p><td align="left" width="400"><h3>スポーツ界でも注目されている</p> <p>当療術院でも愛用しており、施術中は患者さまにも履いて頂いています。ゴムがきつすぎず、弱すぎず、履きやすいと人気の商品です。
</p> <p>こちらは女性用：短・長 1足 400円。日本製。</h3></td></p>
<p><td align="left" width="200" height="180" align="center"></td></p>	<p><td align="left" width="400"><h3>こちらが男性用です。
</p> <p>各色 1足 400円。日本製。</h3></td></p>
<p><td align="left" width="200" height="180" align="center"></td></p>	<p><td width="400" align="left"><h3>“血液サラサラ”効果で人気上昇中のダッタンそば茶！
</p> <p>当療術院で以前から飲んでおり、毎回施術後に患者様にお出ししています。
</p> <p>くせがなく飲みやすく、そばの香りもふわっと。ノンカフェインですので子供からお年寄りまで安心してお飲みいただけます。
</p> <p>100g400円にて販売しております。</h3></td></p>

<table border="0" style="width: 800px; margin: auto;"> <tr> <td colspan="2" style="text-align: center;"><table align="center" width="800" border="0" cellspacing="0" cellpadding="0"></td> </tr> <tr> <td colspan="2" style="text-align: center;"><tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"><td></td></td> </tr> <tr> <td colspan="2" style="text-align: center;"></tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"><tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"><td height="35" align="left" valign="top">ご購入方法</td></td> </tr> <tr> <td colspan="2" style="text-align: center;"></tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"><tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"><td align="left" valign="top"></td></td> </tr> <tr> <td colspan="2" style="text-align: center;"></tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"></table></td> </tr> </table>		<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">		<tr>		<td></td>		</tr>		<tr>		<td height="35" align="left" valign="top">ご購入方法</td>		</tr>		<tr>		<td align="left" valign="top"></td>		</tr>		</table>	
<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">																							
<tr>																							
<td></td>																							
</tr>																							
<tr>																							
<td height="35" align="left" valign="top">ご購入方法</td>																							
</tr>																							
<tr>																							
<td align="left" valign="top"></td>																							
</tr>																							
</table>																							

<table border="0" style="width: 800px; margin: auto;"> <tr> <td colspan="2" style="text-align: center;"><table align="center" width="800" border="0" cellspacing="0" cellpadding="0"></td> </tr> <tr> <td colspan="2" style="text-align: center;"><tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"><td align="left"><h3>インターネットでのご購入は現在行っておりませんので、ご来院の際にご購入下さい。</h3></td></td> </tr> <tr> <td colspan="2" style="text-align: center;"></tr></td> </tr> <tr> <td colspan="2" style="text-align: center;"></table></td> </tr> </table>		<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">		<tr>		<td align="left"><h3>インターネットでのご購入は現在行っておりませんので、ご来院の際にご購入下さい。</h3></td>		</tr>		</table>	
<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">											
<tr>											
<td align="left"><h3>インターネットでのご購入は現在行っておりませんので、ご来院の際にご購入下さい。</h3></td>											
</tr>											
</table>											

<!-- 下部メニューバー -->

</table>

<table width="800" align="center">

<tr>

<td width="800"></td>

</tr>

<tr>

<td align="center">| トップページ


```

</a>|<a href="intro.html"> 療術院紹介 </a>|
      <a href="access.html"> 来院方法 </a>|<a
href="products.html"> 商品紹介 </a>|<a href="reserve.html"> ご予約・お問い合わせ
</a>|</td>
      </tr>
    </table>
</body>
</html>

```

[reserve.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html>
<head>
  <META http-equiv="Content-Style-Type" content="text/css">
  <title>たかくさき療術院 -ご予約・お問い合わせ</title>
  <META name="Author" content="たかくさき療術院">
  <META name="Description" content="たかくさき療術院のホームページです。こち
らは予約・お問い合わせのページです。当院では機械を使用せずに、夫婦の手だけでマッサージ
をするので、子供から高齢者まで、また妊娠の方などでも誰でもマッサージを受けることができ
ます。日々蓄積される体の凝りを解消し、体本来の治癒力を高め、頭痛、腰痛、肩こりから抜け
毛、生理不順、不妊などの症状を改善しましょう。">
  <META name="keywords" content="たかくさき, たかくさき療術院,療術院,マッサー
ジ,神奈川県,横浜,本郷台, 自然治癒力, 体のメンテナンス, 4つの手,抜け毛,頭痛,肩こり,腰痛,不
妊,出張">
  <META name="robots" content="index,nofollow">
  <style type="text/css">
  <!--
#ImgContainer {height:110px;}
img{border:none;}
-->
</style>
</head>

<script type="text/javascript">
<!--

var img1_on = new Image();
var img1_off = new Image();
var img2_on = new Image();
var img2_off = new Image();
var img3_on = new Image();
var img3_off = new Image();
var img4_on = new Image();
var img4_off = new Image();
var img5_on = new Image();
var img5_off = new Image();
img1_on.src="images/top-button02.gif";
img1_off.src="images/top-button03.gif";
img2_on.src="images/intro-button02.gif";
img2_off.src="images/intro-button03.gif";
img3_on.src="images/access-button02.gif";

```

```

img3_off.src="images/access-button03.gif";
img4_on.src="images/product-button02.gif";
img4_off.src="images/product-button03.gif";
img5_on.src="images/reserve-button02.gif";
img5_off.src="images/reserve-button03.gif";

function rollon(targetimage){

    if(targetimage == "image1"){
        document.images[targetimage].src=img1_on.src;

    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_on.src;

    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_on.src;

    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_on.src;

    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_on.src;

    }
}

function rolloff(targetimage){

    if(targetimage == "image1"){
        document.images[targetimage].src=img1_off.src;

    }else if(targetimage == "image2"){
        document.images[targetimage].src=img2_off.src;

    }else if(targetimage == "image3"){
        document.images[targetimage].src=img3_off.src;

    }else if(targetimage == "image4"){
        document.images[targetimage].src=img4_off.src;

    }else if(targetimage == "image5"){
        document.images[targetimage].src=img5_off.src;

    }
}

</-->
</script>
<body bgcolor="#FFFF99" background="images/back-image.gif" text="#000000"
link="#CC6633" vlink="#CC6633" alink="#CC6633" topmargin="0" leftmargin="0"
marginwidth="0" marginheight="0">
<SCRIPT LANGUAGE="JAVASCRIPT"
SRC="http://j1.ax.xrea.com/l.j?id=100412602"></SCRIPT>
<NOSCRIPT><A HREF="http://w1.ax.xrea.com/c.f?id=100412602"

```

```

TARGET="_blank"><IMG SRC="http://w1.ax.xrea.com/l.f?id=100412602&url=X"
BORDER="0"></A></NOSCRIPT>
<!-- バナー -->
<table width="800" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td width="800" height="100" colspan="5"><IMG
src="images/banner02.gif"> </td>
  </tr>
</table>

<!-- 上部のメニューバー -->
<table align="center" width="800" height="40" border="0" cellspacing="0"
cellpadding="0" background="images/opt-bar02.gif">
  <tr>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="131" height="40" align="center"><a
href="index.html"><IMG name="image1" src="images/top-button03.gif"
onMouseOut="rolloff('image1')" onMouseOver="rollon('image1')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="131" height="40" align="center"><a
href=intro.html><IMG name="image2" src="images/intro-button03.gif" alt="療術院紹介"
onMouseOut="rolloff('image2')" onMouseOver="rollon('image2')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="121" height="40" align="center"><a
href=access.html><IMG name="image3" src="images/access-button03.gif" alt="来院方法"
onMouseOut="rolloff('image3')" onMouseOver="rollon('image3')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="121" height="40" align="center"><a
href=products.html><IMG name="image4" src="images/product-button03.gif" alt="商品紹介"
onMouseOut="rolloff('image4')" onMouseOver="rollon('image4')"></a></td>
    <td width="15" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
    <td width="170" height="40" align="center"><a
href=reserve.html><IMG name="image5" src="images/reserve-button03.gif" alt="予約・お問
い合わせ" width="170" onMouseOut="rolloff('image5')"
onMouseOver="rollon('image5')"></a></td>
    <td width="21" height="40" align="center"><IMG
src="images/bar-spacer03.gif" height="25"></td>
  </tr>
</table>

<!--本文-->
<table align="center" width="800" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><IMG src="images/spacer.gif"></td>
  </tr>
  <tr>
    <td height="35" align="left" valign="top"><IMG
src="images/title-sideicon.gif"><font size="6">ご予約方法・お問い合わせ</font></td>
  </tr>

```

```

        <tr>
            <td align="left" valign="top"><IMG src="images/dottedline.gif"
width="300"></td>
        </tr>
        <tr>
            <td width="500" align="left" cellspacing="0" cellpadding="0"
style="padding-bottom:20px;"><font size="5">お電話にてご予約・お問い合わせ下さい。
<br><br>
            ・予約状況は下記の『予約状況閲覧』にてご覧いただけます。<br>
            ・出張も承ります。<br>
            ・その他お気軽にご相談下さい。</font></td>
        </tr>
        <tr>
            <td width="500" align="left" cellspacing="0"
cellpadding="0"><font size="6">tel&fax 045(894)6713</font></td>
        </tr>
        <tr>
            <td height="35" align="left" valign="top"
style="padding-top:20px;"><IMG src="images/title-sideicon.gif"><font size="6">予約状況閲覧</font></td>
        </tr>
        <tr>
            <td align="left" valign="top"><IMG src="images/dottedline.gif"
width="300"></td>
        </tr>
        <tr>
            <td width="500" align="left" cellspacing="0" cellpadding="0"
style="padding-bottom:20px;"><font size="5">現在の療術院の予約状況を閲覧できます。空いている時間帯をご確認のうえご予約ください。</font></td>
        </tr>
        <tr>
            <td width="500" align="left" cellspacing="0" cellpadding="0"
style="padding-bottom:20px;"><font size="5"><a href="reserve/pc/yoyaku_joukyou.php">予約状況を閲覧</a></font></td>
        </tr>
    </table>
</body>
</html>

```

2. 進捗報告資料

4hands

2006年10月12日(木)

PM 江木典之
環境3年 安藤亮一
環境3年 向吉学
環境2年 真野恵理子

今週やったこと

- 質問項目の用意
 - ヒアリング・10月11日(水)
-

質問項目

- 現在の予約方法
 - 来客層
 - 現在のインフラ
 - gooを見て来た客数
 - 遅れるという連絡
-

ヒアリング内容

- 現状の予約方法
 - 電話、メールでのやりとり
 - 空いてる時間を客がいちいち尋ねる
 - 高草木さんが紙の表に書き込む
 - 携帯にも予約の時間を入力している

 - 一人あたりの時間が定まっていない
 - 表に直接書き込む人もいる(松澤さん)
-

ヒアリング内容

- 来客層
 - 学生から、お婆さんまで
 - お年寄りでもインターネットを利用している
-

ヒアリング内容

- インフラ
 - ノートパソコンはある
 - インターネットはこの機会に導入してもいいと思っている
 - 現状は携帯と紙の表を使って管理している
-

ヒアリング内容

- gooを見て来た客数
 - gooを見て問い合わせはあるが実際には来ない
 - 実際に来る人は口コミで広がる
-

ヒアリング内容

- 遅れるという連絡
 - 頻度はほとんどない
 - 一人あたりの時間は基本的に1時間30分
 - 余分に前後に30分ずつとっている
-

ヒアリング結果

- 予約システムというより予約表を共有
 - アナログの使い勝手を保つ
 - たかくさき夫妻が主に携帯で予約表の管理ができるもの
-

今後のタスク

- 利用者にもヒアリングする
 - ヒアリング結果から作るべきものを考える
 - 高草木さんに提案する(ヒアリング2回目)
-

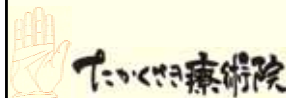
たかくさ療術院

4hands 進捗発表

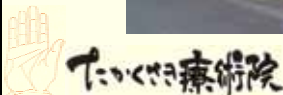
- PM 江木 典之
- 環境情報3年 向吉 学
- 環境情報3年 安藤 亮一
- 環境情報2年 真野 恵理子

今週行ったこと

- ユーザアンケートのお願い
- 予約過程のロールプレイ
- ユースケース作成
- システム範囲の策定

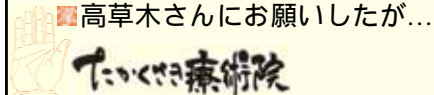


再びたかくさ療術院へ



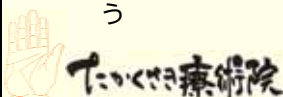
ユーザアンケートのお願い

- 対象
 - 療術院にくるお客様
- 目的
 - 携帯・PCでインターネットを利用する人はどの程度か
 - 実際に予定表の閲覧をしたいと思うか



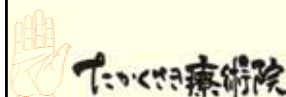
アンケート手法の問題

- アンケートの説明に関して
 - お客様に対しての説明になってない
- 質問項目に関して
 - フォーマットが自由すぎて返答しづらい
- 集計方法に関して
 - 集計しても、ただ意見の羅列になってしまう



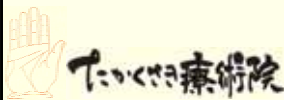
再びユーザアンケートのお願い

- 高草木さんからの手法の改善のアドバイス
 - 集計・返答のしやすさなどを考えて
 - 各項目に関して選択肢を設けたらどうか？
 - 携帯所有者とPC所有者で別々に質問項目を設けたらどうか？



予約過程のロールプレイ

- 実際に予約をいれてみた
 1. お互いに自分の予定表を見ながらやり取り
 2. 2、3度のやり取りの末、予約
 3. 予約表に予約状況を書き込み
 4. 更に携帯にも予約状況を入力

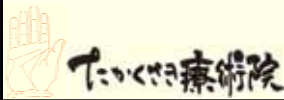


実際にやってもらった



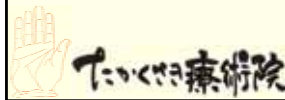
ヒアリングとロールプレイから わかった問題点

- 電話等でやり取りしないと予約の空き状況がわからない
- 紙の予定表と携帯のスケジュール、両方に予約情報を入力している



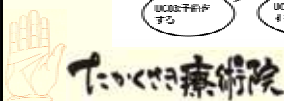
ヒアリングからわかった要望

- 予約システムというより予約表を共有
- アナログ（現状の予約表）の使い勝手を保つ
- 高草木さんが主に携帯で予約表の管理をできるもの



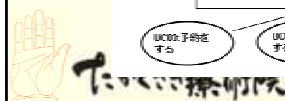
ユースケース作成

- システムの全体構想を明らかにするため、ユースケースを作成



システム範囲の策定

- 高草木さんの要望、現状の問題点を踏まえて、システム化範囲を策定



来週以降の課題

■画面仕様書

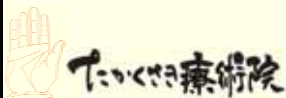
■画面のレイアウト

■データの入出力 etc

■アーキテクチャ仕様書

■どんなソフトウェアを使うか

■何のDBを使うか etc



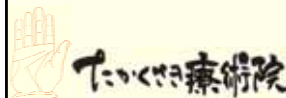
たかくま治療院

4hands 進捗発表

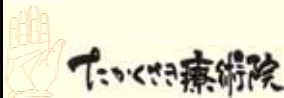
- PM 江木 典之
- 環境情報3年 向吉 学
- 環境情報3年 安藤 亮一
- 環境情報2年 真野 恵理子

報告内容

- 全体のスケジュール
- 今週の作業
- 成果物
- 今後の課題

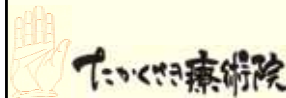


全体スケジュール



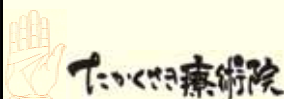
今週の作業

- 作業時間
- ミーティング 2 hr
- 安藤 6 hr
- 向吉 3 hr
- 真野 3 hr



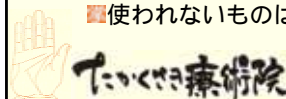
今週の作業

- 作業内容
- アンケートの修正・再依頼
- 提案書のドラフト作成
- 技術検証

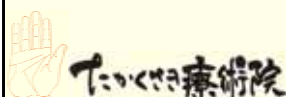


アンケートの修正・再依頼

- アンケートの修正
- 選択方式にする
- 手紙とアンケートの部分を別ける
- アンケートの目的
- 本当にお客様が利用してくれるのか
- 使われないものは作りたくない

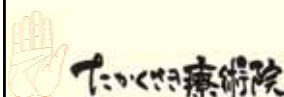


修正版アンケート用紙

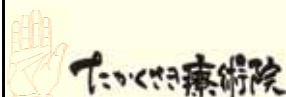


提案書ドラフト作成

- 次のミーティングで推敲
- 次週中にたかくさきさんへ

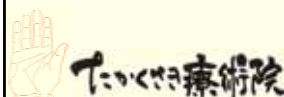


提案書ドラフト版

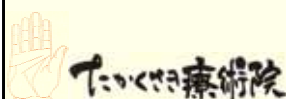


技術検証

- フリー予約ソフトの導入
- ソフトウェア構成の検討
 - CGI、DB、システム構成
 - メリット・デメリット

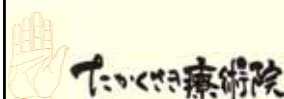


技術アセスメント



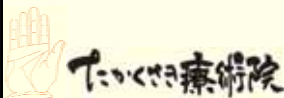
成果物

- 修正版アンケート
- 提案書のドラフト
- 技術アセスメント

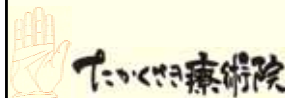


今後の課題

- 画面仕様書の作成
- プロトタイプ作成
- 紙と携帯の二重入力をどうするか
- 紙のアナログ感をどうするか
 - DS ワイヤレスネットワークのインフラ前提



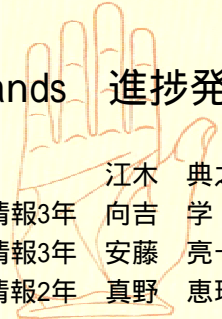
ご清聴ありがとうございました。



たかくさ療術院

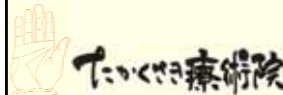
4hands 進捗発表

- PM 江木 典之
- 環境情報3年 向吉 学
- 環境情報3年 安藤 亮一
- 環境情報2年 真野 恵理子



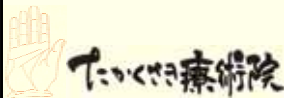
報告内容

- 全体のスケジュール
- 今週の作業
- 成果物
- 今後の作業



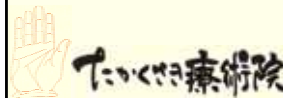
全体スケジュール

10月	11月	12月	1月	2月
方向付け	推敲	作成	移行	
・ヒアリング ・ユースケース ・提案書 ・技術検証	・画面仕様書 ・プロトタイプ			



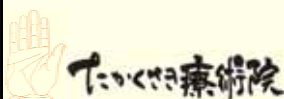
今週の作業

- 作業時間
- ミーティング 2 hr
- 安藤 6 hr
- 向吉 7 hr
- 真野 6 hr



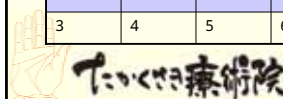
今週の作業

- 作業内容
- プロトタイプの作成
- 画面仕様書の作成
- たかくささん訪問（画面仕様書の承認）



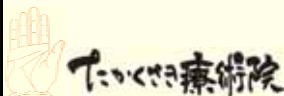
今週の作業

日	月	火	水	木	金	土
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
画面仕様書の作成		訪問		プロトタイプの作成		--
26	27	28	29	30	12/1	2
	--	--				
3	4	5	6	7	8	9



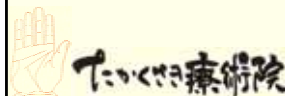
成果物

- 画面仕様書
- プロトタイプ（未完成）



プロトタイプ

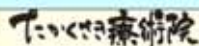
- 参照画面
 - ブラウザを利用
- 更新画面
 - 携帯アプリを採用
 - 携帯ブラウザは表現力・機能に限界、文字が小さい、操作や入力が困難



画面遷移図（予約状況の更新）



備考



画面仕様書

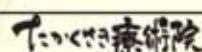
画面ID: PO103 画面名: 予約入力画面

画面イメージ



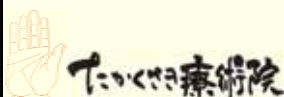
仕様

- 登録ボタンの押下時に、以下のチェックを行う。
 - 入力の日付であるか
 - 日付ではない場合、メッセージを表示する
 - 入力が必要な項目であるか
 - 日付の場合、予約状況画面に遷移する



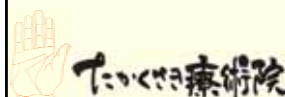
たかくさきさん訪問

- 操作・機能の考えの食い違いがないように明確に。
- たかくさきさんの要望も聞き、プロトタイプに反映する。
- 画面仕様書の承認をいただきました

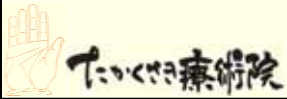


今後の作業

- プロトタイプの完成
- プロトタイプのテスト
 - 予定より少し遅れをとっているので追いつきたい

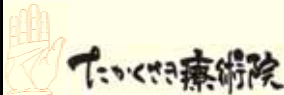


ご清聴ありがとうございました。

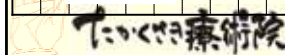
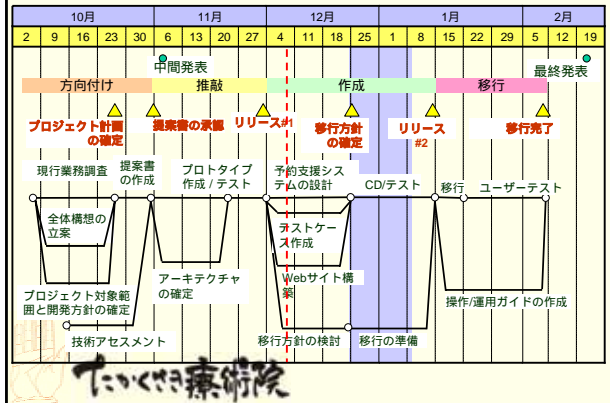


4hands 進捗報告

- Agenda
 - マスタースケジュール
 - 進捗上の問題と対応
 - 今週の作業進捗状況(予定と実績)

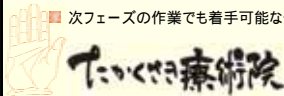


マスタースケジュール



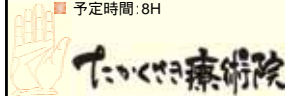
進捗上の問題点とその対応 (先週)

- 11月30日の状況
 - 大学祭のための休暇もあり進捗が遅れていた。11月末がマイルストーンであった推敲フェーズの完了(リリース#1)が遅延している。
 - 特に、携帯アプリとサーバー間でのリクエスト送受信の実装方法が未確定
- 11月30日にチーム内での検討と対応
 - スケジュールの確認
 - サービスインから逆算した場合の、スケジュール感(デッドライン)を共有
 - 今後の作業を再確認し、担当作業の見直しを確認
 - 冬休みにコーディング/テストを可能にする 12/21までにフィードバックをもらう必要あり 12/14にリリース#1
 - リリース#1は、メインの処理が行えることを対象とする
 - 項目チェック、エラー処理は対象外とする
 - 目的は、アーキテクチャの確定、お客様に使用してもらいフィードバックをいただくこと
- 次フェーズの作業でも着手可能な作業は実施する



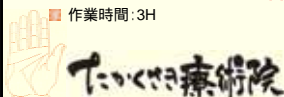
今週の作業予定

- 【向吉さん】
 - 予約状況入力画面(携帯アプリ)の実装
 - 携帯アプリと各サーバーとの通信の実装
 - 予定時間: 20h
- 【安藤さん】
 - 携帯アプリと各サーバーとの通信の実装
 - サーバー側の予約状況データの保持方法の検討
 - データを読み込み部分の実装
 - 予定時間: 10H
- 【真野さん】
 - 「UC05: サイト構築」のコンテンツの検討
 - レイアウトのドラフトの作成
 - 高草木様へのレビュー実施
 - 予定時間: 8H



今週の作業進捗状況

- 【向吉さん】
 - 予約状況入力画面(携帯アプリ)の実装 **済** 実装済み、(UTは未完)
 - 携帯アプリと各サーバーとの通信の実装 **済**
 - 作業時間: 15h
- 【安藤さん】
 - 携帯アプリと各サーバーとの通信の実装 **済**
 - サーバー側の予約状況データの保持方法の検討 **済**
 - データを読み込み部分の実装 **済** 実装済み、(UTは未完)
 - 作業時間: 12H
- 【真野さん】
 - 「UC05: サイト構築」のコンテンツの検討 **実施中**
 - レイアウトのドラフトの作成 **実施中**
 - 高草木様へのレビュー実施 **本日コール予定** 体調不良のため、作業時間が少なめ。
 - 作業時間: 3H



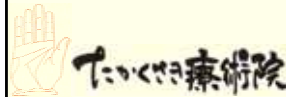
たかくま治療院

4hands 進捗発表

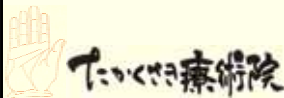
- PM 江木 典之
- 環境情報3年 向吉 学
- 環境情報3年 安藤 亮一
- 環境情報2年 真野 恵理子

報告内容

- スケジュール
- 今週の進捗
- 作業時間
- 今後の作業

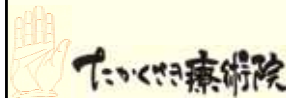


スケジュール



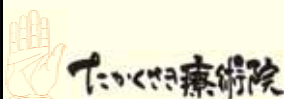
今週の進捗

- プロトタイプ作成・テスト
- 高草木さん訪問（リリース#1）
- サイト構築



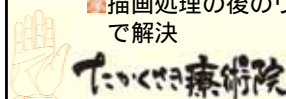
デモ

- 感想、コメントがあったら評価の紙に書いてください



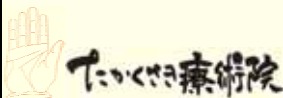
プロトタイプ作成・テスト

- 実機でテストしたら動かなかった
 - 特定の操作ではなく色々な場面で強制終了
 - メモリ関連か（Nではメモリ関連にバグ）
 - エラーメッセージを見ると画面関連
 - 計算と描画のCPUが別で描画処理が溜まっていく（Nでは）
 - 描画処理の後のウェイト時間を延ばすことで解決



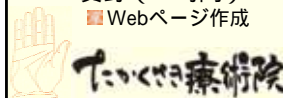
高草木さん訪問

- 携帯にアプリをダウンロードして頂き、操作方法を説明した
- 「いいね、わかりやすい」と好感触



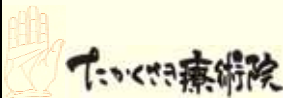
作業時間

- 安藤 (9 時間)
 - プロトタイプテスト : 3hr
 - たかくさきさん訪問 : 2hr
 - 環境の整備 : 4hr
- 向吉 (13 時間)
 - 実装 : 5hr
 - テスト : 1hr
 - 修正 : 5hr
 - たかくさきさん訪問 : 2hr
- 真野 (4 時間)
 - Webページ作成



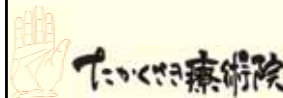
今後の作業

- リリース # 1 を使っていたいただいた感想を伺う (来週水曜日)
- 作成フェーズの開発計画の作成



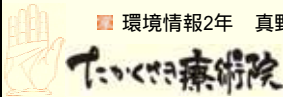
サーバについて

- 運用して頂くことになった場合レンタルサーバを考えていた
- 大岩研のサーバを使ってもいいですか？



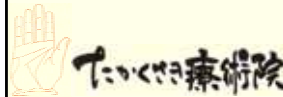
4hands 進捗発表

- PM 江木 典之
- 環境情報3年 向吉 学
- 環境情報3年 安藤 亮一
- 環境情報2年 真野 恵理子



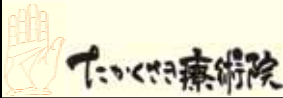
報告内容

- 全体のスケジュール
- 今週の作業
- ミーティング
- 今後の課題



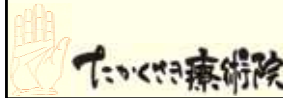
全体スケジュール

- WBS



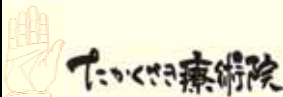
今週の作業

- 作業時間
 - Webサイト構成の策定
 - 安藤 2hr
 - Webサイトデザイン案
 - 向吉 7hr
 - 真野 6hr
 - ミーティング 2hr



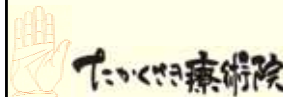
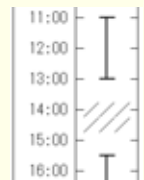
今週の作業

- Webサイト構成の策定
 - どういったコンテンツを含めるか
 - アウトラインを作成
- Webサイトデザイン案
 - たかくさぎさんと話し合って土台を作成
 - いくつかのデザイン案を用意
 - いい部分をマージしていく



ミーティング@ドトール湘南台

- リリース1(試作版)を使ってみて
 - 予約の閲覧
 - 斜線と実線がごちゃごちゃ
 - 特に携帯で見ると...
 - 線や休みの日の説明が無い
 - 予約の管理
 - 1週間の表での表示は好評
 - 携帯での閲覧でもこの表を使えないか?
 - 入力面でインタフェースの改良の余地あり
 - 違った入力方法を提案



ミーティング@ドトール湘南台

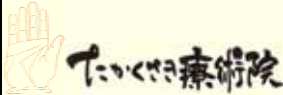
■ リリース2(製品版)にむけて

■ 予約の閲覧

- 用事(斜線)と予約(実線)を統一
携帯週別表のように色で表現
- 携帯での見え方に工夫を
携帯週別表のような表現

■ 予約の管理

- 用事(灰色)と予約(青)を統一
- 入力方法の刷新
開始時間と終了時間を表から選択する形式で入力



ミーティング@ドトール湘南台

■ リリース2(製品版)にむけて

■ エンドユーザからの要望

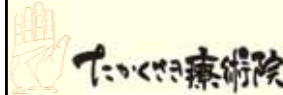
- 個人情報のため、ユーザ認証が必要
- ID・パスワードの管理

自分の予約した時間
を確認したい

インターネットで
予約したい

方針

- リリース範囲には入れない
- 1. 期間的な問題
- 2. 現状のシステムの質を上げたい
- 3. まずは情報の発信から

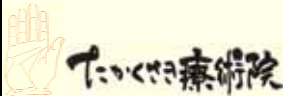


今後の作業

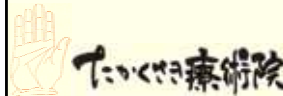
■ 作成フェーズ

■ リリース2(製品版)の作成

- 予約の閲覧(PC/携帯)
- 予約の管理
- Webサイト



ご清聴ありがとうございました。



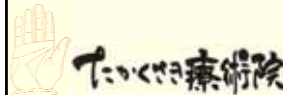
たかくさ療術院

4hands 進捗発表

- PM 江木 典之
- 環境情報3年 向吉 学
- 環境情報3年 安藤 亮一
- 環境情報2年 真野 恵理子

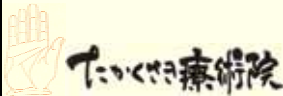
報告内容

- 全体のスケジュール
- 冬休み中の作業
- 成果物
- 今後の作業



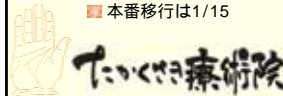
全体スケジュール

- WBS参照



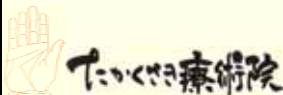
冬休み中の作業

- 12/21のミーティングで決めた内容
- 担当
 - 携帯アプリ - 向吉
 - 文字入力ではない入力方法への変更
 - チェック項目の定義・サーバへ送るデータの定義
 - ブラウザ - 安藤
 - 線・斜線の変更
 - GDの使用 サーバにインストール
 - HP - 真野
 - メニューバーを上部へ
 - 作成
- 冬休み前にたかくささんを訪問する
- 画面仕様書の提出・HPデザインの確認
- 1/11までに
 - 合意したものを完成させる（統合テスト済み）
 - HTMLもアップ
- 本番移行は1/15



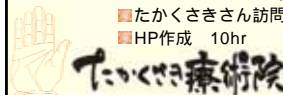
冬休み中の作業

- 12/28 たかくささん訪問
 - 画面仕様書の提出・HPデザインの確認
- 12/29 SKYPE会議
 - 現状の進捗報告
 - たかくささんと合意した内容の報告



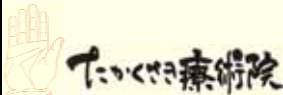
冬休み中の作業

- 作業時間
- 安藤 15hr
 - たかくささん訪問 1hr
 - 実装・テスト 14hr
- 向吉 16hr
 - 画面仕様書の作成 3hr
 - たかくささん訪問 1hr
 - 実装・テスト 12hr
- 真野 14hr
 - HPのテンプレート作成 3hr
 - たかくささん訪問 1hr
 - HP作成 10hr



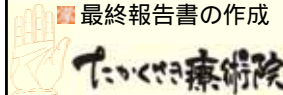
成果物

- 携帯アプリ
- ブラウザ
- ホームページ



今後の作業

- 携帯アプリ・ブラウザ
 - 統合テスト
 - たかくさきさんによるシステムテスト
 - 操作ガイド・運用ガイドの作成
- HP
 - HPの内容を詰める
 - HTMLをアップする
 - たかくさきさんに評価してもらう
- 最終報告書の作成



ご清聴ありがとうございました。

