システム・ソフトウェア開発

自由制作レポート

宿泊施設の客室状況管理アプリ

氏名: 木越　湧太

1. **作品概要**

私は本演習において「システム・ソフトウェア開発」を選択し, ビジネスホテル等の宿泊施設での使用を想定した「客室状況管理アプリ」を作成した. 以降, 本作品と記す.

1. **開発のきっかけ**

私は2年ほどビジネスホテルの客室清掃のアルバイトをしており, その職場での「不便」を解消しようと思い, システム・ソフトウェア開発による解決策の提案を考えた. 具体的な不便を感じる点として, 勤務先のビジネスホテルでは紙媒体の清掃記録用紙を使用している点がある. 1つの客室の清掃開始時と終了時にその都度, 清掃者の名前と時間を記入しなければならないのに加えて, 各スタッフがどの部屋を清掃しているのかなどの連絡がLINEグループでやりとりされ, 手間がかかる. さらに忘れ物を発見した時や設備不良等の問題が発生したときにもLINEグループで写真を添付して報告する必要がある. これらの作業が非効率的に感じ, 部屋状況の確認や報告等は同じアプリケーションから行うことで手間を減らし, 的確な指示のみがLINEグループでされることで清掃の効率化を図ることができると考え, 本作品の開発を決定した.

1. **開発環境と想定**

本作品の開発環境はAndroid Studioで, 言語はJavaを使用した. 対象の端末として, 「Google Nexus9」というタブレットを対象としており, エミュレータもこれを使用して開発した. 私が勤務しているビジネスホテルでの使用を想定しており, 客室は2階から9階に各フロア15部屋ずつあることを前提に開発した.

1. **本作品の画面構成**

本作品は7つの画面から構成される. 画面構成図を図1に示す. 実行時に表示されるメイン画面では作業内容を, 「客室管理」「設備不良リスト表示」「忘れ物リスト表示」の3つから選択できる. メイン画面表示の様子を図2に示す. 「客室管理」のボタンを押すと, フロア選択画面に遷移する. フロア選択画面を図3に示す. この画面でフロアを選択すると選択されたフロアのフロア画面(客室選択画面)に遷移する. フロア画面を図4に示す. フロア画面では15個の客室ボタンが配置されており, そのボタンを押すと選択された客室の客室設定画面に遷移する. 図5に客室設定画面を示す. その画面中の「House Keeping」ボタンを押すと, 客室清掃設定画面に遷移し, 清掃の記録ができる. 客室清掃設定画面を図6に示す. メイン画面の「設備不良リスト表示」ボタンを押すと, 設備不良リスト表示画面に遷移する. 設備不良リスト表示画面を図7に示す. 同様に「忘れ物リスト表示」ボタンを押すと, 忘れ物リスト表示画面に遷移する. 忘れ物リスト表示画面を図8に示す.

ダイアグラム

自動的に生成された説明

図1 画面構成図

コンピューターの画面

自動的に生成された説明

図2 メイン画面

コンピューターの画面

自動的に生成された説明

図3 フロア選択画面

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

図4 フロア画面

パソコンの画面のスクリーンショット

自動的に生成された説明

図5 客室設定画面

パソコンの画面のスクリーンショット

自動的に生成された説明

図6 客室清掃設定画面

コンピューターのスクリーンショット

自動的に生成された説明

図7 設備不良リスト表示画面

パソコンの画面

自動的に生成された説明

図8 忘れ物リスト表示画面

1. **アプリケーションの機能**

アプリケーションの機能は以下に示す(1)から(3)の3つに分けられる.

1. 客室のステータス変更

客室の状態は主に, 空室, 客がチェックインしている状態, 客がチェックアウトして汚れている状態, 清掃中, 清掃が終了して清掃チェックが必要な状態に分けられる. これらの状態を本作品ではアイコンとして客室ボタンに表示する設定にした. これらの客室状態アイコンと客室状態フローを図９に示す. 本作品では各客室設定画面から「Checked In」「Checked Out」のボタンと, 客室清掃設定画面の「Cleaning Start」「Cleaning Finish」「Inspection Start」「Inspection Finish」ボタンが押されたときに, アイコン表示を切り替える. 図10は213号室の客室設定画面において「Checked In」ボタンが押された直後のフロア画面での213号室のアイコン表示の変更を表す図である.

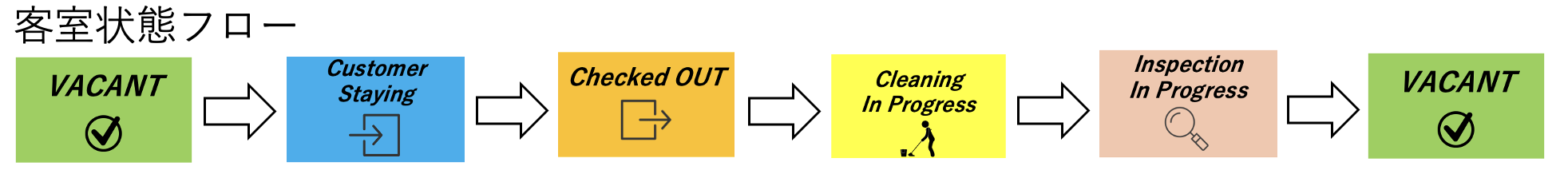


図9 客室状態アイコンと客室状態フロー

グラフィカル ユーザー インターフェイス, アプリケーション, Teams

自動的に生成された説明

図10 213号室でのステータス変更例 (Vacant→Customer Staying)

1. 清掃記録

客室清掃設定画面において清掃者の名前, 清掃開始時間, 終了時間を記録することができるのに加えて, 客室清掃ができているかのチェックである「Inspection」の開始と終了に関しても記録でき, 押された場合はアイコン表示を更新する. 図11は213号室の客室清掃設定画面においてTomさんが清掃を終えた時の実行例である.

パソコンの画面のスクリーンショット

自動的に生成された説明

図11 213号室での清掃設定例

1. 忘れ物・設備不良登録と一覧表示

客室清掃設定画面においてテキスト入力欄に忘れ物または設備不良内容を入力し, 「Report」ボタンを押すと登録が完了する. 登録された情報は, メイン画面から忘れ物リスト表示または設備不良リスト表示に移ると, 部屋番号, 入力された時の日付, 内容を1行に表示される. このリストでは追加された順に上から表示される. 図12は213号室で発見された財布の忘れ物を一覧表示画面で表示している様子を示した図である.

パソコンの画面

自動的に生成された説明

図12 忘れ物の一覧表示例

1. **制作中に生じた課題・苦労した点と解決策**

客室数は1フロア15部屋×8階分(2階から9階) = 120部屋あり, これらのすべての客室設定画面を作成しなければならないということに気づき, どのように選択した客室の固有の情報を引き継ぎつつ, 120部屋分の客室設定画面を1つの画面で汎用して実装するかという点が, 最も苦労した点であった. 各客室の客室清掃設定画面においても同様の課題が発生した.

選択したボタンの客室情報は, 選択された客室番号をIntentで引き継ぎ, 遷移先画面でIntentから取得することで遷移先の画面で選択された客室に対する操作を行うことができると考え, 実装してみた. しかし, 全客室の固有の情報をそれぞれ管理することが新たに課題として生じたので, 部屋番号と客室ステータスをフィールドとして持つRoomクラスを定義し, そのRoomクラスの配列をフィールドとして持つRoomManagementクラスをさらに定義して遷移先の画面でRoomManagementクラスのインスタンスから, 選択された (操作を行いたい) 客室を取得し, ステータス変更等の操作を行うことで, 画面遷移と客室管理を分離して実現することができた. これにより, 120部屋分の客室設定画面と客室清掃設定画面, 2階から9階の8階分のフロア画面を1つの画面設計で汎用化することに成功した.

1. **本作品のプログラム概要**

本作品ではjava.androidにあるApplicationクラスを継承したMyHotelApplicationというアプリケーション実行を担当するクラスを作成し, その中で部屋管理用のRoomManagementクラスを動かしている. 図13は客室管理を実現している本作品のアプリケーション構造図である. リスト1は本作品で使用している「一括管理されている客室情報の取得と変更」のプログラム例で, 213号室の客室をRoomManagementクラスから取得し, 清掃中に変更する例である. この例に示すプログラムに加えて本作品ではIntentから取得した部屋番号に依存させ, ステータスによるアイコン表示変更, 画面表示設定の変更, ボタン設定の変更を行っている.

ダイアグラム

自動的に生成された説明

図13 アプリケーション構造図

リスト1 客室情報の取得と変更のプログラム例

1 MyHotelApplication myApp = getApplication();

2 RoomManagement rm = myApp.getRoomManagement();

3 int roomNumber = 13;

4 int floor = 2;

5 Room room213 = rm.getRoomInformation(roomNumber, floor);

6 room213.changeStatus( “Cleaning In Progress” );

1. **考察と今後の展望**

考察として, 本作品制作を通してMyHotelApplicationクラスの中でRoomManagementクラスを動かし, RoomManagementクラスが各部屋のステータスを保持し, その参照方法はRoomクラスのメソッドが担当するといった, オブジェクト指向における「カプセル化」と「処理の分担」について再確認できた. また身の回りの生活の中から清掃記録の不便さという「問題」を発見し, その問題に対してどのような仕様のソフトウェアにするかという仕様設計, そして設計をもとに開発し, 開発中に生じた課題に対する解決策の模索といった社会課題解決のプロセスを自分なりにトレースできたと考える.

今後の展望として, ボタン配置や使用感の調整などUI面についてこだわりを入れること, さらにはリネンやアメニティの在庫状況確認の機能を追加するとさらに良い仕様になると考える. また, 実際に勤務先での使用を考えると, 複数の端末から同じデータにアクセスして使用できるといった機能追加も要求されると考える.