

これで、想定したタスクに合ったUIデザインができました。

ところが実際にアプリケーションを作ってみると、いくつか追加の要求が出てきました。たとえば、次のようなことです。

- 本の情報の入力間違いを修正する
- 本を処分した場合にその登録を削除する
- 読み終わった本の感想文を書き込む
- すべての本をリストにしてざっと見る
- 本を著者別や出版社別に並べて見る

もしこれらの新たな要求を満たすとすると、UIは図1-1-2のように冗長になっていきます。やること別に入口が並んでいるだけの、いかにもアマチュアライクなデザインです。

問題は、このUIがユーザーの目当てを現していないことです。目当てが見えていないので、タスクだけが羅列される形となって、道具としてのまとまりに欠けたものになっています。このアプリケーションが何の情報を扱うもので、その全体像はどのようになっているのかが、はっきりしないのです。それは設計者であるあなたの頭の中だけに暗黙的に想定されているにすぎません。ここでは、アプリケーションが前提としている問題とその解法が、利用者に対する恣意的な行動統制として一方的に決定されているのです。ではその目当てとは何のことでしょうか。

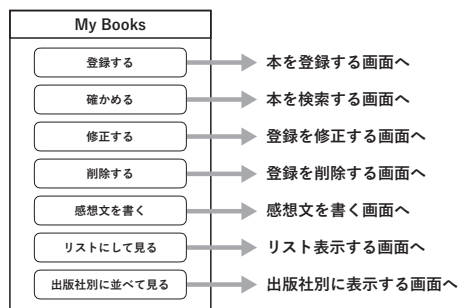


図1-1-2 新たな要求を満たしたUI

目当て=オブジェクト

先述の課題は、オブジェクト指向でUIを作ることによって解決できます。目当て=オブジェクトを起点としてUIをデザインするのです。この場合のあなたの目当ては「本」です。本の情報を登録したりそれを検索したりするのがアプリケーションに想定されたタスクですが、それらのタスクの対象となっているもの、あなたが関心を寄せているそのものは、要するに「本」なのです。ですから、オブジェクト指向UIでは、ユーザーに「本」の存在を見せることが表現上の中心的な活動になります。そのUIは図1-1-3のようになるでしょう。

このUIでは、あなたの蔵書がまず一覧表示されます。まるで自分の本棚を見ているようです。あなたの関心の対象である「本」がそのまま画面の中にある。これがオブジェクト指向UIなのです。そしてそこで本を検索したり、一冊を選んで内容を確認／編集したり、新しい本を登録したりできるようにするのです。

あとでくわしく解説しますが、スマートフォンやパソコンの画面で私たちが

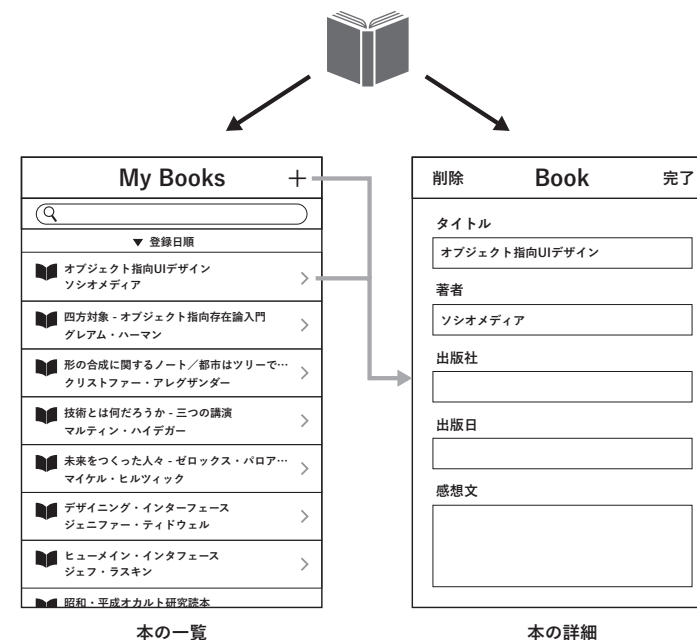


図1-1-3 オブジェクト指向によるUI

3-3 ステップ1. オブジェクトの抽出

まずはオブジェクトの抽出です。

次の図は、学校名簿アプリケーションをイメージしたタスクのサンプルです。断片的で粒度もまちまちなものですが、アプリケーションが発想されたときの小さな目的意識の例として挙げています。

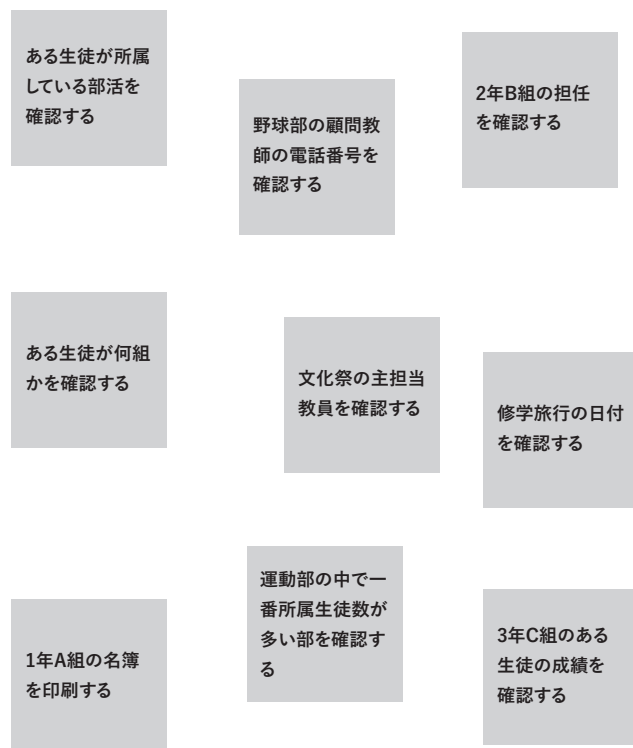


図3-3-1 学校名簿アプリケーションのタスク

想定されたタスクとして書かれた単語や表現は、アプリケーションのデザインに大きな影響を与えます。実際のデザインでは事前のリサーチや要求分析などによってこのタスクを揃えるところから始まります。ここではたとえばこう

いったタスクがぼんやりと考えられ、ひとまずこれをさっと形にしてみるならば、という状況を考えてください。

このタスクの断片を例にしながら、オブジェクト指向UIのアプローチ方法を説明していきます。

「名詞」を抽出する

それぞれのタスクの文言から名詞を探します。

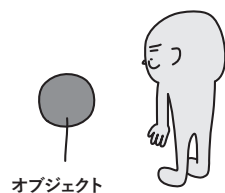


図3-3-2 タスクから名詞を抽出する

オブジェクトは名詞を手掛かりに抽出されます。その際、単に名詞を機械的に抜き出すのではなく、ユーザーの関心の対象となる概念は何かを考えてみて

ください。

オブジェクトには「もの」的なものもあれば、「こと」的なものもあります。食べ物や書物などもオブジェクトですし、食事や読書などもオブジェクトです。



オブジェクト

「もの」	
○	＝
○	＝
○	＝
○	＝
○	＝



アクション

「もの」	
○	＝ (もの)
○	＝ (もの)
○	＝ (もの)
○	＝ (もの)
○	＝ (もの)



オブジェクト化されたアクション

「こと」	
⑦	＝
⑦	＝
⑦	＝
⑦	＝
⑦	＝

図3-3-3 オブジェクトには「もの」的なものも「こと」的なものもある

探してみるとあれもこれもオブジェクト、日々の生活では普段からたくさん
の名詞に囲まれています。24時間365日、私たちはオブジェクトの中で暮ら
しているのです。

「名詞」とそれらの関係を抽出する

次に名詞同士の関係を整理します。

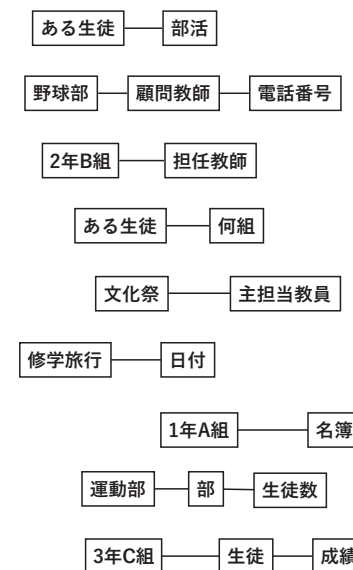


図3-3-4 名詞同士の関係を抽出する

タスクごとに名詞同士の関連を示すために線でつなげていきましょう。このあたりはUMLのクラス図を使ってシステムが扱う基本概念をモデリングした経験があれば慣れた作業だと思います。



図レベル4-7 モバイル

さらに発展させるとどうなるでしょうか。

今回のタスクにはありませんでしたが、ユーザーにとって「スペースが今すぐ使えるかどうか」が特に重要かもしれません。そのような場合はスペースのコレクションビューにおいてフィルタリングを用意しましょう。

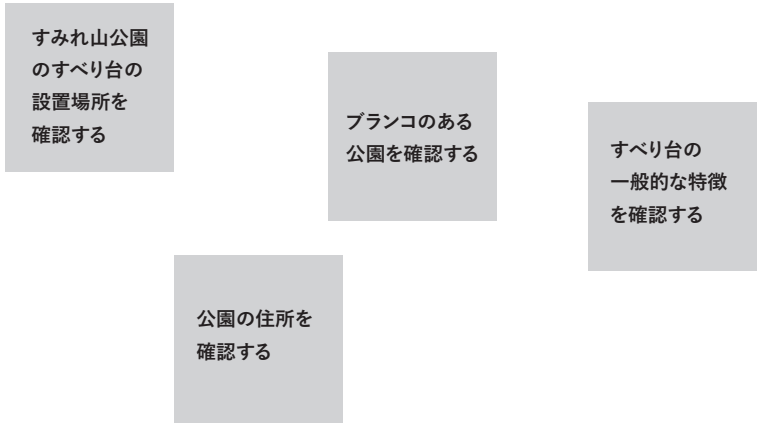
フィルタリングについては後のワークアウト (レベル9) で扱っているので参考にしてください。

なお、このワークアウトでは触れませんでしたが、実際にこのような予約システムを作る上では「予約をしたユーザー」を特定するために「ユーザー」オブジェクトが必要になるでしょう。

想定タスクの文言には表れていなくても、実装寄りの観点からモデリングをしたい場合には、そのような一般的に必要とされる重要なオブジェクトを加えて全体の概念構成を検討していてもかまいません。

レベル5 家族で遊べる場所を探すアプリケーション

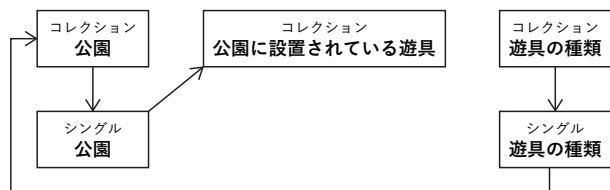
次のタスクから、家族で遊べる場所を探すアプリケーションをデザインしてください。



図レベル5-1 家族で遊べる場所を探すアプリケーションのタスク

ビューとナビゲーションの検討

次はコレクションビューとシングルビューの呼び出し関係を整理します。



図レベル5-3 ビューとナビゲーションの検討

公園に設置されている遊具は、情報量が少ないのでシングルビューは省略しています（レイアウト面積に依存するのでレイアウトしながら決めましょう）。また公園シングルビューから遊具の種類コレクションビューへのナビゲーションは不要と考え省略しました。

ルートナビゲーションには「公園」と「遊具の種類」を並べることにしました。ただし「遊具の種類」という表現は少し複雑な印象を与えるので、ここでは単に「遊具」としてみましよう。

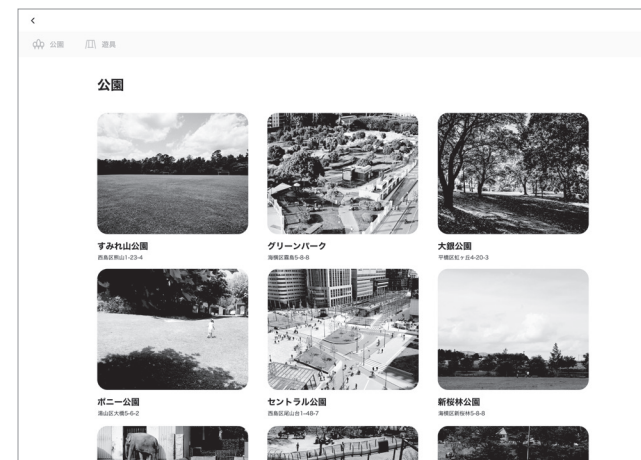
レイアウトパターンの適用

レイアウトパターンを適用して実際の画面をデザインしていきます。

次の図はデスクトップのサンプルです。

公園のシングルビューには、設置されている遊具のコレクションビューを一体化して表示しています。プロパティは遊具の名称と設置場所が表示されています。実際には設置されている各遊具に固有の名称がない場合も多いかもしれないので、名称だけでなく遊具の種類と関連付けるようにオブジェクトをモデリングしてもよいでしょう。

公園のプロパティに定義されていない写真を大きく使っています。「公園の写真を確認する」というタスクは言語化されにくいものです。ですが写真を使ってレイアウトすると具体的なイメージを掴むのに有用であることがわかります。このような場合はオブジェクトのプロパティに写真を追加しましょう。このようにレイアウトからオブジェクトを定義するという方向もあり得ます。



公園のコレクションビュー



公園のシングルビュー

図レベル5-4 デスクトップ