

Designing your CA app

Susan McKeever



Intro

- Diving straight from the “idea” to coding is a bad idea
- Need to articulate your design:
 - What it does
 - How it does it
 - What it looks like -
- Your design enables someone to develop it
 - good way

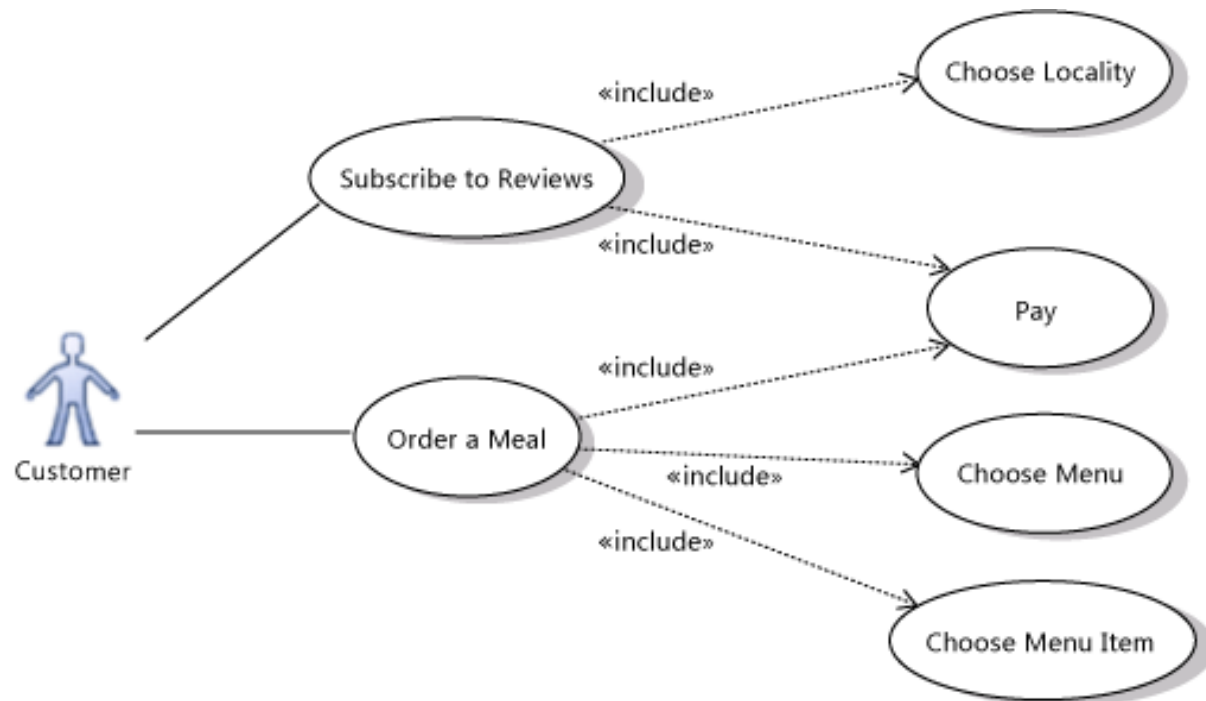
Specifics:

- What it does
 - Functionality
 - Design Artefacts: Use cases; Description of what the app does; screen flows
- How it does it
 - Technical info:
 - Design Artefacts: Class diagrams; Database ERD, Tech arch (if cloud)
- What it looks like
 - Design artefacts: Screen designs (inc. navigation – check existing app)

Usability of your app

- Nielsen's 10 usability heuristics still apply
- +
- Minimise data entry - never ever get a user to type in data **unnecessarily** (you will need some data entry). E.g. drop down boxes are easier than typing full value;
 - Avoid overcrowding the screen
 - Get someone else to check it's self explanatory
 - Adopt sensible navigation (look at existing apps)

Reminder:





Use case

Use case

Functions

<<include>>



<<extend>>



Relationships

A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



Actor

Users

Design spec should contain

- Use cases;
- Description of what the app does;
- Screen flows
- Class diagrams; Database ERD
- Tech arch diagram if including cloud
- Screen designs (inc. navigation – check existing app)
- If any of it is hand drawn, make sure they are legible.

2nd Monday Nov – firstname_surname.zip
submission in webcourses

Marking

- No separate grade (feedback)
- If not done, multiplier of 0.8 used on app assignment
- The right way to approach a user application