

CONCEPT
DOCUMENTATIE
COOK ie

Interaction Design - Kernmodule 3

Ivy Barend

3012556

Cookie Cutter Recipes



For any starting Chef !

CONCEPT

COOKie is een handige online applicatie die iedere beginnende Chef kan helpen met het bereiden van een lekkere maaltijd. Het helpt overzicht creëren, de kook-paniek te verminderen, en zo tijds-efficiënt mogelijk je gerecht te bereiden. Alle recepten komen uit een openSource database, waar jij als gebruiker ook je eigen recepten kan formatteren en toevoegen. Zo leer je samen met alle andere beginnende Chefs de subtiële kunst van koken op een rustige en overzichtelijke manier!

Aanleiding en Proces

Ik kwam op het idee voor COOKie vanwege mijn eigen frustratie bij sommige recepten-Sites. Op dat punt was ik al een tijdje in de leer bij mijn vriend en betere Chef, die het op zijn beurt weer heeft geleerd door honderdduizend youtube-filmpjes. Maar ik kon me nog duidelijk herinneren, de eerste keer dat ik op kamers ging, en ik niet eens precies wist wanneer een aardappel precies klaar was met koken. Dit resulteerde in een jaar lang belabberd eten, met voorverpakte Maggi-kruiden en waterige soep. Later stapte ik over op het volgen van recepten die ik van internet heb afgeplukt, maar daar zit een groot probleem in: de menselijke invloed.

Veel mensen die al goed zijn in iets, hebben de neiging om de 'obvious' stappen over te slaan. Maar wat betekent het 'fruiten' van een ui? Hoeveel tijd kost het om een goede saus te laten indikken? Al deze technieken en begrippen waren niet duidelijk voor me. Daarbij kwam nog dat ik niet efficiënt was met mijn tijd, waardoor een recept van 40 minuten me 2 uur kostte. Deze dingen wilde ik oplossen, door een simpel, makkelijk te volgen format aan te bieden die al deze dingen opving.

De uitdagingen die ik omtrent dit concept voor mezelf heb gesteld, zijn als volgt:

- Efficiëntie
- Overzichtelijkheid
- Laagdrempeligheid

Mid-presentatie

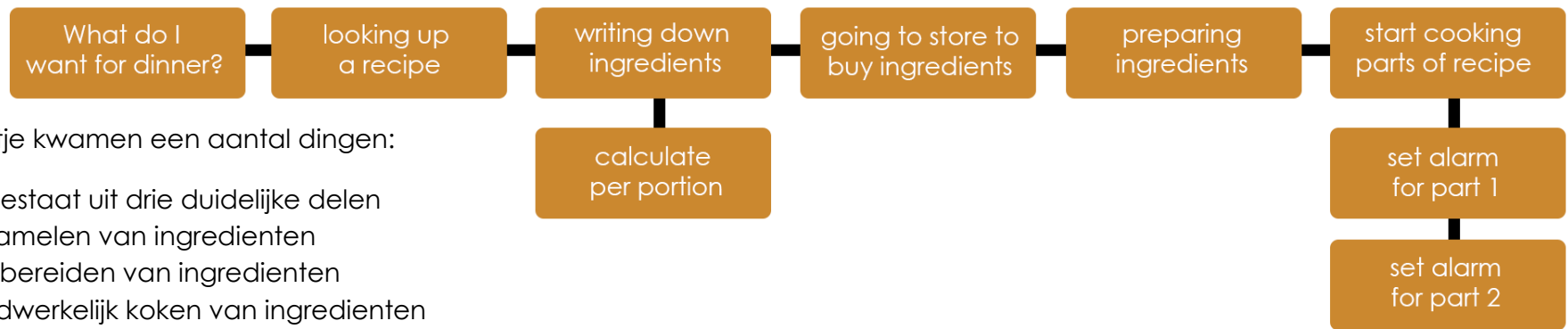
Mijn concept wat ik presenteerde tijdens de mid-presentatie, had een vrij hakkelige flow:



Ik had hier alle stappen al in stukken gehakt, met onderscheid tussen de lijst met ingrediënten en de daadwerkelijke uitvoering. De applicatie liet de ingrediënten zien, zodat je die in alle rust kan verzamelen, en daarna loopt hij eigenlijk van begin tot eind door alle individuele stappen, tot je klaar bent met koken. Het idee er achter was dat als ik de gebruiker aan het handje meeneem, en één stap per keer laat zien, ik de efficiëntie neerzet waar ik naar streefde, en daarmee de overzichtelijkheid.

De feedback die ik op dit concept ontving, was dat er juist een soort paniekerig gevoel voortkomt uit het één voor één laten zien van al die stappen. De gebruiker moet dan de hele tijd zijn apparaat in de gaten houden, tegelijkertijd met zijn gasfornuis. Dit reflecteerde op mijn eigen ervaring met conventionele recepten-sites, waarin ik wel 5 keer per stap check of ik het goed heb gedaan, besepte ik dat dit inderdaad niet the way to go was. Verder had dit concept ook geen overzichtelijke representatie van de data, wat tenslotte de kern van de opdracht was.

Back to the drawing board: ik moest op de een of andere manier een balans vinden tussen overzichtelijkheid en efficiëntie, en dat komt neer op de hoeveelheid informatie die ik per keer aan de gebruiker laat zien. Hierop besloot ik even een korte Customer Journey op te zetten omtrent de actie van koken, nog los van mijn concept.



Uit dit overzichtje kwamen een aantal dingen:

- Koken bestaat uit drie duidelijke delen
 - Verzamelen van ingrediënten
 - Voorbereiden van ingrediënten
 - Daadwerkelijk koken van ingrediënten
- De aangegeven hoeveelheden op receptensites zijn niet altijd wat de gebruiker nodig heeft
- Het ingewikkeldste onderdeel van koken bestaat uit het bijhouden van de tijd voor verschillende onderdelen
- De grootste paniek en het grootste obstakel voor efficiëntie komt uit het bijhouden van tijd

Hieruit kon ik verder met mijn concept. Ik besloot de stappen die je per recept verloopt in drie schermen te verdelen: de ingrediënten van het recept, het voorbereiden van die ingrediënten, en het bereiden van het gerecht. Dat laatste zal het belangrijkste scherm zijn, waar de meeste dynamische informatie wordt gegeven. De uitdaging was dus ook om dat scherm, met de data omtrent kooktijden, zo overzichtelijk mogelijk te visualiseren.

Op dit punt leek het me de juiste stap om een recept op te gaan zoeken, zodat ik aan de hand van dat recept mijn ontwerp kan uitdenken en finetunen. Ik besloot te gaan voor een recept waar veel informatie en stappen bij kwamen, maar waarbij ik wel duidelijk kon laten zien dat met de juiste visualisatie, het heel simpel kan zijn. Ik ging voor chef John (*from foodwishes.com!*) 's Meatball Marinara.

Links naar het recept:

- <https://www.allrecipes.com/recipe/220854/chef-johns-italian-meatballs/>
- <https://www.allrecipes.com/recipe/11966/best-marinara-sauce-yet/>

ONTWERP

Vanuit hier ging ik mijn schermen uitdenken. Ik heb flink lopen schetsen in mijn ideeënboekje. De eerste 2 schermen zijn statische informatie, dus daar lag niet de grootste uitdaging. Het laatste scherm heb ik wel even mee lopen worstelen.

Scherm 1: het boodschappenlijstje

Functionaliteit

Scherm één moet heel simpel de benodigde ingrediënten voor het recept weergeven. Sommige ingrediënten worden verdubbeld in de verschillende onderdelen, maar die moeten wel bij elkaar opgeteld worden weergegeven op dit scherm. Verder, aan de hand van mijn bevindingen omtrent user experience, wilde ik ook een functionaliteit die de hoeveelheden per portie kan verminderen of vermeerderen. De gebruiker kookt namelijk niet altijd voor de hoeveelheid mensen die de schrijver van het recept had bedacht.

Vormgeving

Qua vormgeving van dit scherm heb ik gekozen voor een klassiek boodschappenlijstje. Dit heb ik gedaan omdat het herkenbaar en overzichtelijk is. Het geeft een beetje een air van vertrouwdheid aan een nieuwe vorm van koken, en het is algemeen genoeg om aantrekkelijk te zijn voor een brede doelgroep.



Scherm 2: Mise en Place

Functionaliteit

De informatie die moest worden weergegeven in dit scherm is al een stuk minder statisch. Dit scherm moest de verschillende stappen van de voorbereiding weergeven aan de gebruiker, en aan welke stap een timer hangt, moet worden gestart wanneer de stap doorlopen is. De stappen moeten op de meest efficiënte manier worden doorlopen, en in de juiste volgorde, en de gebruiker moet kunnen zien welke stappen al doorlopen zijn.

In mijn eerste ontwerp heb ik me vooral gefocussed op hoe de gebruiker weet welke stap hij al doorlopen heeft. Ik besloot hierom dit scherm ook op te delen in verschillende stappen. Ten eerste de Mise en Place, dan de voorbereiding van de gehaktballen, en dan de voorbereiding van de saus. Vervolgens kan de gebruiker ergens op drukken om aan te geven dat de stap doorlopen is, en als alle stappen per onderdeel verlopen zijn, springt het vak op "gedaan" en kun je verder met de volgende. Als er een timer aan de stap hangt, zal deze aanspringen op het moment dat de gebruiker op 'volgende stap' drukt.

Vormgeving

Zoals je kunt zien in de eerste afbeelding hiernaast, heb ik de functionaliteit vrij letterlijk vertaald naar de vormgeving. Ik was vanwege de feedback die ik tijdens de Mid-presentatie had gekregen, een beetje angstig om de informatie los weer te geven, in de vorm van één stap per keer op je scherm. Echter, na het Post

Mise en Place

- soak breadcrumbs in milk for 20 minutes

- ✓ grate the parmesan
- ✓ dice the onions
- ✓ mince the garlic
- chop the parsley

- glaze half of the diced onions in a pan for about five minutes

Meatballs

mix together in a bowl:

- 500 g ground pork
- 500 g ground beef
- breadcrumb mix
- 2 eggs
- 2 tbs fresh parsley
- 3 cloves of garlic
- 2 ts salt
- 0.5 ts black pepper
- 0.5 ts oregano
- 2 tbs parmesan
- glazed onion

- cover bowl and refrigerate for about an hour

- roll mixture into balls of about 4 cm
- lay them out on a baking sheet

Marinara

grind together with a food processor:

- stewed tomatoes
- tomato paste
- 4 tbs fresh parsley
- 1 clove of garlic
- 1 ts oregano
- 0.25 ts black pepper
- 1 ts salt

2

mix the following ingredients together in a bowl:

- 500 g ground pork
- 400 g ground beef
- breadcrumb mix
- 2 eggs
- 2 tbs fresh parsley
- 3 cloves of garlic
- 2 ts salt



Mortem kreeg ik de feedback dat te veel interactie met de applicatie niet goed is, omdat je tijdens het koken simpelweg vieze handen hebt en je niet voor elke scheet aan je telefoon of tablet wilt zitten. Dit forceerde me om de vormgeving een beetje om te denken. Uiteindelijk besepte ik me ook dat het format wat ik had bedacht niet op ieder recept toepasbaar was, en ik een algemener systeem moest bedenken.

Uiteindelijk besloot ik de individuele stappen van de Mise en Place niet op te delen in 'categoriën', zoals in mijn eerste ontwerp, maar gewoon simpelweg per stap een blokje informatie met de instructie, de benodigde ingrediënten en een foto weer te geven. Wanneer de gebruiker klaar is met de stap, moet hij wel op het blokje drukken om naar de volgende te gaan, maar hiermee is de interactie met het scherm toch al veel minder dan in het eerste ontwerp. Verder heeft de gebruiker wel de mogelijkheid om de eerdere stappen terug te bekijken, voor het geval hij wil checken of alles wel goed gegaan is, maar de actie is optioneel en het zorgt er voor dat de visuele informatie per stap geminimaliseerd wordt.

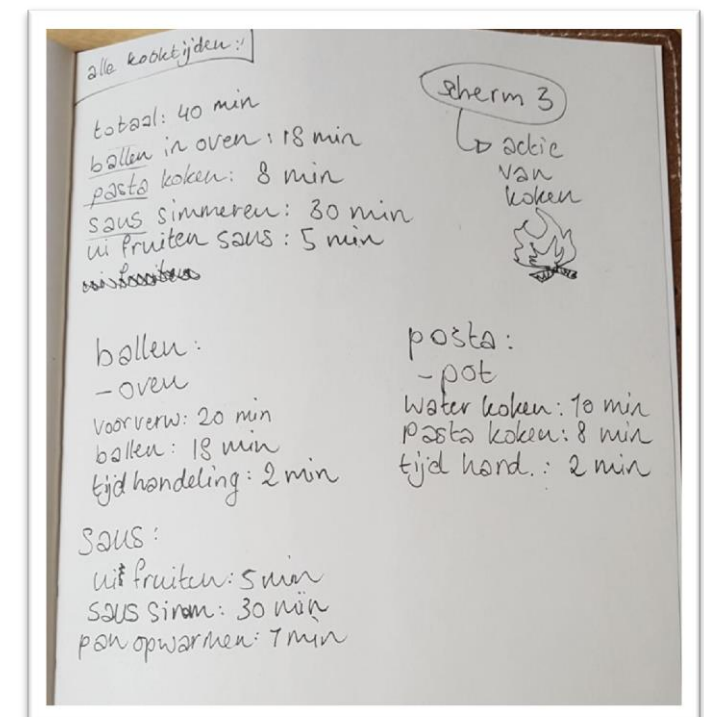
De timer verplaatst zich naar bovenaan het scherm, met het stapnummer, zodat de gebruiker nog wel altijd kan zien hoe lang een stap nog duurt.

Scherm 3: het koken

functionaliteit

Ik moest het proces van koken omvormen tot een overzichtelijke hulp. Ik wilde dus alle onderdelen tegelijkertijd in beeld hebben, zonder dat het hektisch wordt. Tijdens het brainstormen ben ik begonnen met alle kooktijden opschrijven, en daarin viel me op dat ik de kooktijden in 3 stukken kon hakken: alles rondom de oven, alles rondom de sauspan en rondom de pasta pan. Dit leek me een goed concept om op door te bouwen, dus daar ging ik mee aan de slag.

Al schetsend probeerde ik een visueel verantwoorde, overzichtelijke manier te vinden om die onderdelen te representeren. Om alles zo efficiënt mogelijk te houden, rekende ik de totale kooktijd uit en stelde de losse onderdelen tegenover de totale kooktijd. Als ik dat allemaal tegelijkertijd zou aflopen, heeft de gebruiker een totaaloverzicht van wat er



allemaal gaande is op één plek, en dat moest de paniek van allemaal losse kookwekkers wegnemen.

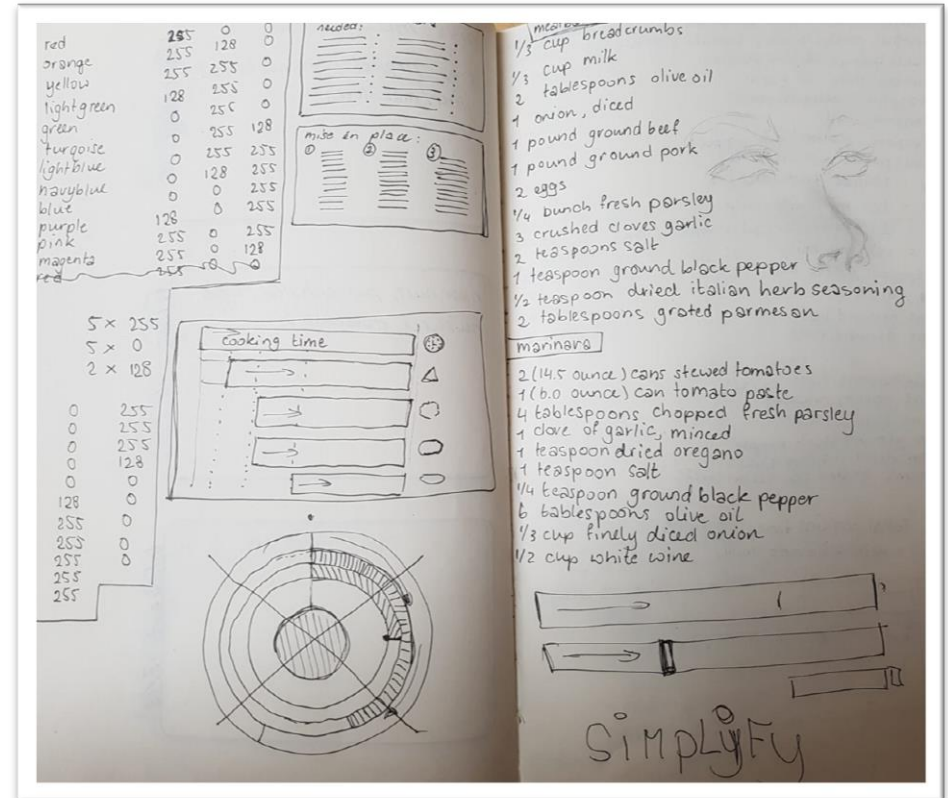
Verder moeten de instructies bij de stappen binnen een onderdeel nog worden weergegeven. Er komt een notificatie boven de relevante balk te staan, wanneer deze ingaat. De notificatie blijft voor 20 seconden staan en verdwijnt daarna, maar hij kan weer worden opgeroepen door de balk aan te klikken.

Vormgeving

Ik moest dus een startpunt en een eindpunt hebben, en alle losse onderdelen moesten tegelijkertijd klaar zijn. Zoals je kunt zien in mijn schetsen heb ik nog even gespeeld met het idee om het als een ronde vorm weer te geven, maar omdat dat toch te veel het gevoel opwekt van een aflopende klok, en tegelijkertijd weinig ruimte overlaat voor indications welke aflopende klok voor welk onderdeel was, ben ik begonnen met de vrij simpele maar wel efficiëntste en meest overzichtelijke versie die ik kon bedenken: een stel rechthoeken. Echter lopen ze wel van links naar rechts af, met als gezamenlijk eindpunt het eindpunt van de totale kooktijd.

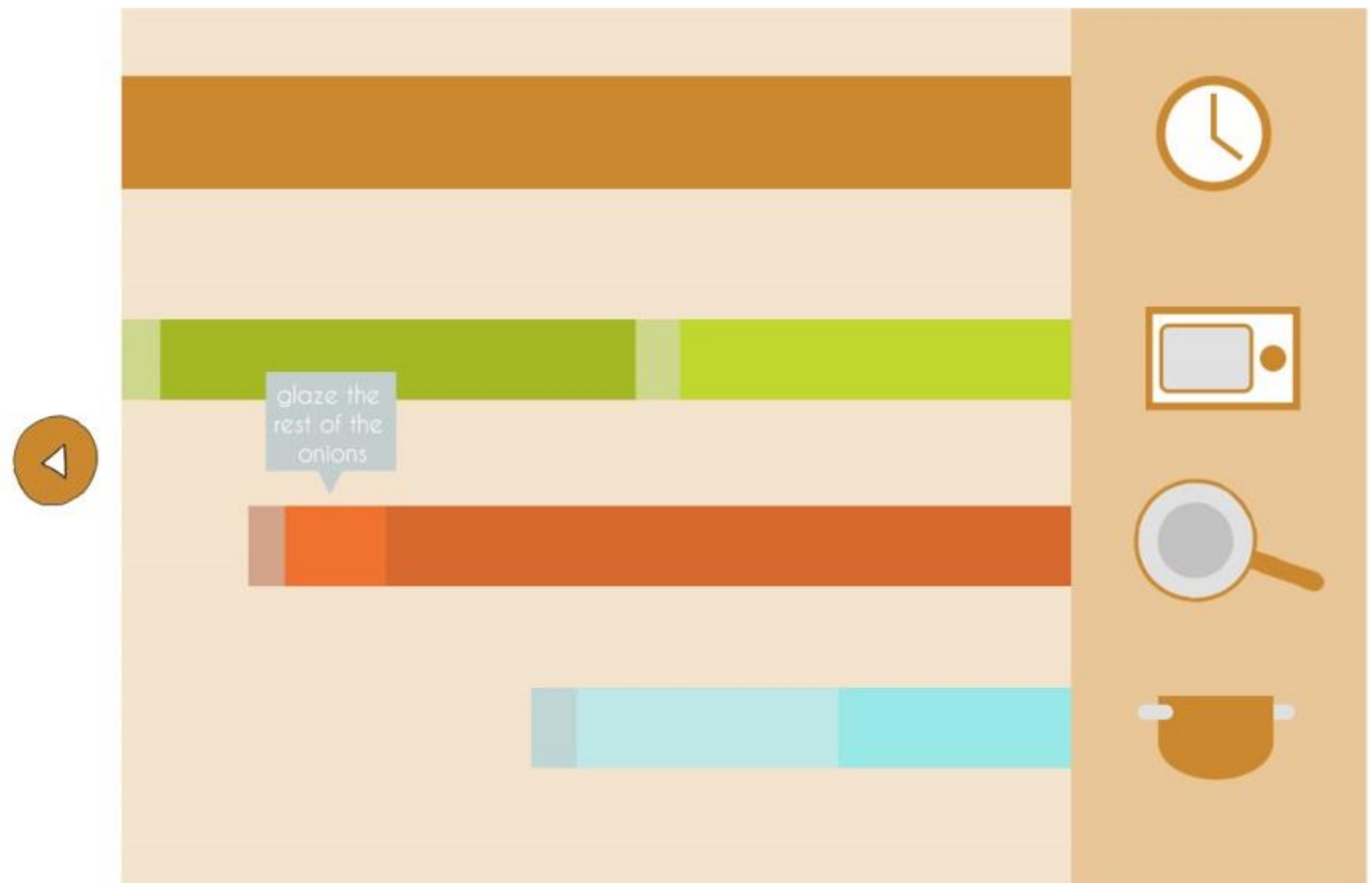
Het was ook nog een optie om alle onderdelen in één aflopende rechthoek te stoppen, maar dat vond ik weer te veel informatie achter elkaar. Als je aan het koken bent, wordt je aandacht verdeeld over de verschillende onderdelen van je keuken, dus de pannen/oven. Het leek me dus aansluitend op de user experience het meest logisch om ook die onderdelen apart te visualiseren.

Verder qua kleurkeuze ben ik in de verschillende onderdelen gegaan voor een verschillende kleur per balk, maar binnen de balk wel dezelfde tinten. Hierdoor is duidelijk dat iedere actie binnen een onderdeel met hetzelfde 'punt' in de keuken te maken heeft, maar het maakt een subtiel onderscheid tussen de verschillende onderdelen. De kleuren representeren min of meer de onderdelen, waarbij groen voor de oven omdat de gehaktballetjes door de verse peterselie een beetje groenig zijn, rood voor de sauspan omdat de marinara-saus ook rood is, en blauw voor de pastapot omdat de pasta in water kookt.



De totale tijd-balk is in de huistint van oranje, dit om aan te geven dat het de algemene kooktijd is en de andere onderdelen samenvat.

Alle onderdelen worden nogmaals aangegeven in de vorm van iconen, zodat het voor de gebruiker makkelijk is om de punten in zijn keuken te hangen aan de tijd-balken per onderdeel.



IDEALE VORM

Functionaliteit

COOKie zou worden uitgewerkt als een downloadbare applicatie voor op je smartphone of tablet. De gebruiker kan zowel recepten toevoegen aan de database, als er een uitkiezen en mee koken.

Het toevoegen van een recept wordt in een per-stap format gedaan. Aan de hand van de informatie die de gebruiker per stap aangeeft, namelijk de ingrediënten, of er een timer aan zit en hoe lang alles duurt, kan de applicatie zelf bijvoorbeeld de totale unieke ingrediënten uitrekenen, en het derde kookwekkerscherf inrichten.

Wanneer de gebruiker een recept uitkiest, wordt er een beginscherf getoond waar de kooktijden en de moeilijkheidsgraad van het recept worden getoond. Vanaf dat de gebruiker op start drukt, wordt de interactie zo veel mogelijk geminimaliseerd.

Het eerste scherm wat je te zien krijgt is het boodschappenlijstje. Wanneer de gebruiker er klaar voor is kan hij naar links swipen waarna de voorbereidingsstappen worden getoond. De gebruiker moet per stap eenmaal op de box klikken om door te gaan naar de volgende stap. De gebruiker heeft de mogelijkheid om terug te scrollen naar de vorige stappen, maar wanneer de gebruiker het scherm 'loslaat' zal hij automatisch terugspringen naar de huidige stap. Om het scherm te 'locken' op een stap die al gedaan is, moet de gebruiker snel nogmaals op de box tappen.



Wanneer alle instructies voorbij zijn en alle timers bij die instructies afgelopen, gaat de applicatie automatisch verder naar het laatste scherm. De gebruiker moet hier op start drukken om het kookproces te beginnen. Op deze manier heeft de gebruiker nog even de tijd om alle benodigde keukenspullen er bij te pakken, en te checken of alles wat hij nodig heeft, klaar staat. Wanneer de gebruiker op start drukt, begint het kookproces en wordt de gebruiker begeleid aan de hand van notificaties die omhoog komen als de gebruiker iets moet doen met zijn verschillende 'stations'. De notificatie gaat na ongeveer 20 seconden weer weg, maar kan altijd weer omhoog gehaald worden door op de corresponderende balk te tappen.

Aan het einde van het kookproces gaat de applicatie naar een scherm met een aantal opties. Je kunt foto's van je gemaakte gerecht delen voor andere gebruikers, er staan tips voor het opmaken van je borden en je kunt zien wat anderen er van hebben gebakken! (*pun intended*)

Vormgeving

Voor de huisstijl van COOKie heb ik gekozen voor de tint oranje, omdat het een warme kleur is, maar niet agressief. Het moet een huiselijke sfeer overbrengen en de kleurrijke gerechten mooi omlijsten met een matte kleur. Het lettertype wat ik vrijwel overal gebruik is (heel toepasselijk) Caviar Dreams, dit omdat het leesbaar is, maar niet standaard, en het geeft een beetje een chique Art Deco – tintje aan het hele verhaal.

In het derde scherm maak ik gebruik van tinten corresponderend met het kookstation. De gebruiker die het recept toevoegt zal kunnen aangeven welke keukenapparatuur er wordt gebruikt, en die zal dan in een database corresponderen met een tint.

Cookie Cutter Recipes



PROTOTYPE

Voor het prototype heb ik gekozen om één recept uit te werken, dus alleen de drie schermen die je krijgt wanneer je een recept wilt volgen. De belangrijkste interactie leek me toch het laatste scherm, waar de meest dynamische datavisualisatie plaatsvindt, dus daar heb ik me vooral op gefocussed in de uitwerking.

Scherf 1

Het eerste scherm ziet er zo uit. De ingrediënten worden uit de database gehaald door middel van [deze](#) Query. [Vervolgens](#) wordt er aan de hand van de maximale hoeveelheid ingrediënten, ook uit de database, door de tabel heen geloopt, en worden per column in Vectors gestopt. Dan wordt de Vector per column geprint op het scherm.

```
SELECT ingredient,
      SUM(amount) AS total_amount,
      measure
FROM meatBallMarinara
WHERE ingredientCode = ?
GROUP BY ingredientCode
```

```
if (currentIngredientCode < maxIngredientCode)
{
    currentIngredientCode++;
    //check max amount of unique ingredients
    //add them to vector per column
    recipeQuery->bind(1, currentIngredientCode);
    while (recipeQuery->executeStep())
    {
        ingredient = recipeQuery->getColumn("ingredient").getText();
        ingredients.push_back(ingredient);

        measure = recipeQuery->getColumn("measure").getText();
        measures.push_back(measure);

        amount = recipeQuery->getColumn("total_amount").getDouble();
        amounts.push_back(amount);
    }
    recipeQuery->reset();
}
```

Scherf 2

Het tweede scherm bevat de instructies. De tekst voor de instructies wordt uit de database [gehaald](#) op het moment dat er een nieuw class-object wordt aangemaakt, zoals te zien in [dit](#) stukje code. Het printen van de tekst en ingrediënten wordt gedaan in de draw() functie van de class InstructionBox. Het ophalen van de ingrediënten wordt gedaan door [deze](#) Query.

```
//get instructions from database and put them in class instructionBox
string instructionText = db->execAndGet("SELECT instruction FROM instructions WHERE stepNumber =" + ofToString(instructions.size()));
instruction.setup(listEdge_left, 20, instructionText, instructions.size());
```

```
SELECT ingredient,
      amount,
      measure
FROM instructionIngredients
WHERE stepID = ?
```

```
void instructionBox::update()
{
    if (currentIngredient < ingredientAmount) {
        currentIngredient++;
        instructionsQuery->bind(1, stepNumber);
        while (instructionsQuery->executeStep())
        {
            ingredient = instructionsQuery->getColumn("ingredient").getText();
            ingredients.push_back(ingredient);

            measure = instructionsQuery->getColumn("measure").getText();
            measures.push_back(measure);

            amount = instructionsQuery->getColumn("amount").getDouble();
            amounts.push_back(amount);
        }
        instructionsQuery->reset();
    }
}
```

Scherf 3

Het derde scherm maakt gebruik van kooktijden uit de database. Deze worden **opgehaald** door een hele rits `execAndGet()` statements. Dit omdat het statische data is waar verder niets mee wordt gedaan, het is enkel een referentiepunt voor de aflopende tijdsbalken.

Deze statische data wordt echter wel vermenigvuldigd met de variabele `MinuteMultiplier`. Dit is omdat de kooktijden als minuten in de database staan, en het programma rekent de tijd in seconden. Realistisch gezien moet deze dus 60 zijn, maar hij staat nu op 1 vanwege de demonstratie tijdens het PM.

De tijd in het programma wordt bijgehouden op **deze** manier. Het programma vergelijkt het aantal verlopen seconden sinds de bool 'cooking' true is, en wanneer deze de totale kooktijd bereikt, reset hij zich.

Vervolgens worden de balken op de **volgende** manier opgezet: eerst worden alle balken getekend aan de hand van de individuele kooktijden per stap, met de corresponderende kleuren. Dan wordt daarbovenop een balk getekend wiens breedte aangepast wordt aan de hand van de var 'timeInSeconds'. Op deze manier verloopt de tijd in het scherm. Hier een voorbeeld van de oven.

```
minuteMultiplier = 1;

totalCookingTime = db->execAndGet("SELECT time FROM CookingTimes WHERE tag = 'total'").getInt() * minuteMultiplier;//40, now in se

//cookingtimes for oven
_ovenBar_prep = db->execAndGet("SELECT time FROM CookingTimes WHERE action = 'oven_prep'").getInt() * minuteMultiplier;
_ovenBar_preheat = db->execAndGet("SELECT time FROM CookingTimes WHERE action = 'oven_preheat'").getInt() * minuteMultiplier;
_ovenBar_prep_cook = db->execAndGet("SELECT time FROM CookingTimes WHERE action = 'oven_prep_mb'").getInt() * minuteMultiplier;
_ovenBar_cook = db->execAndGet("SELECT time FROM CookingTimes WHERE action = 'oven_cook_mb'").getInt() * minuteMultiplier;
```

```
if (screen3)
{
    //function for accurately displaying passing of time during cooking
    //check elapsed time in seconds, compare to total cooking time, if reached, reset
    if (cooking == true) {
        timeInSeconds = ofGetElapsedTimef();
        if (timeInSeconds > totalCookingTime) {
            cooking = false;
            ofResetElapsedTimeCounter();
        }
    }
}
```

```
//OVEN

ovenBar_prep.x = totalBar.x;
ovenBar_prep.y = ovenY + 50;
ovenBar_prep.width = ofMap(_ovenBar_prep, 0, totalCookingTime, 0, totalBar.width);
ovenBar_prep.height = totalBar.height;
ofSetColor(ofColor(205, 216, 140)); //pale green
ofDrawRectangle(ovenBar_prep);

ovenBar_preheat.x = ovenBar_prep.x + ovenBar_prep.width;
ovenBar_preheat.y = ovenBar_prep.y;
ovenBar_preheat.width = ofMap(_ovenBar_preheat, 0, totalCookingTime, 0, totalBar.width);
ovenBar_preheat.height = totalBar.height;
ofSetColor(ofColor(192,215,45)); //light green
ofDrawRectangle(ovenBar_preheat);

ovenBar_prep_cook.x = ovenBar_preheat.x + ovenBar_preheat.width;
ovenBar_prep_cook.y = ovenBar_prep.y;
ovenBar_prep_cook.width = ofMap(_ovenBar_prep_cook, 0, totalCookingTime, 0, totalBar.width);
ovenBar_prep_cook.height = totalBar.height;
ofSetColor(ofColor(205, 216, 140)); // pale green
ofDrawRectangle(ovenBar_prep_cook);

ovenBar_cook.x = ovenBar_prep_cook.x + ovenBar_prep_cook.width;
ovenBar_cook.y = ovenBar_prep.y;
ovenBar_cook.width = ofMap(_ovenBar_cook, 0, totalCookingTime, 0, totalBar.width);
ovenBar_cook.height = totalBar.height;
ofSetColor(ofColor(164, 184, 35)); //dark, base green
ofDrawRectangle(ovenBar_cook);

ofSetColor(ofColor(244, 228, 205)); // pale pink
ofDrawRectangle(ovenBar_prep.x, ovenBar_prep.y, timeBar.width, totalBar.height);
```

stewed tomatoes	411 g
tomato paste	170 g
fresh parsley	6 tbs
garlic	4 cloves
oregano	1.5 ts
black pepper	0.75 ts
salt	4 tbs
olive oil	1.1 tbs
onion	2 whole
white wine	125 ml
breadcrumbs	28 g
milk	125 ml
ground beef	500 g
ground pork	500 g
eggs	2 whole
parmesan	2 tbs
linguini	500 g



mix the following ingredients together in a bowl

ground pork
500 g
ground beef
500 g
fresh parsley
6 tbs



PAGINA 1, 2 EN 3 ZOALS
UITEINDELIJK IN HET
PROTOTYPE



add the sauce and white
wine to the skillet and let
simmer, stir occasionally