Table of Contents

[INTRODUZIONE 2](#_Toc507684365)

[Cuculo’s Flight 3](#_Toc507684366)

[Zipped photos 4](#_Toc507684367)

[Domestic zoo 5](#_Toc507684368)

[Business is business 8](#_Toc507684369)

[Super Mario Busy 10](#_Toc507684370)

[Calculator Pro 12](#_Toc507684371)

[Dottor House Dati 13](#_Toc507684372)

[Dottor House Graphics 15](#_Toc507684373)

[Comics Chess 17](#_Toc507684374)

# INTRODUZIONE

Nei capitoli seguenti trovate le specifiche per ogni singolo progetto.

Le regole del progetto:

* Scadenza venerdì 16 marzo (sera);
* Il progetto prevede revisione costante; già lunedì 5 mi aspetto che qualcuno abbia qualcosa da mostrarmi;
* Analizzate il vostro progetto e suddividetelo in task; poi datevi dei tempi (ho diviso in 6 task avendo 12 giorni? Ho 2 giorni per task in media); considerate di dover consegnare due giorni prima della scadenza, così vi restano due giorni di “cuscinetto” se qualcosa va storto in corso d’opera;
* USATE GIT! Lunedì vediamo qualche esempio di utilizzo, così imparate a lavorare in maniera incrementale, e se dopo un po’ incasinate il progetto, sapete come tornare indietro con facilità senza perdere tutto;
* Si possono usare solamente le librerie Bootstrap e jQuery; NON sono permesse altre librerie, e NON è concesso fare copia e incolla brutale di codice pescato a caso da internet; dovete lavorare VOI con i “mattoncini” del web per costruire il sito; se copiate codice senza pensare me ne accorgo;
* Non dovete preoccuparvi di fare un layout *responsive*: preoccupatevi solo che la grafica sia a posto guardando il sito da pc normali con schermi normali;
* Dimenticate il PDF che vi ho già mandato; ho incluso le immagini qui; se tra testo e immagini trovate discordanze, fa fede il testo.
* Per qualsiasi dubbio mandatemi una mail; se vedo che ho indicato un requisito in modo poco chiaro, aggiorno questo documento e lo ricarico sul repository.

# Cuculo’s Flight

Realizzare un videogioco per far volare un cuculo attraverso una foresta.

Il cuculo sta sulla sinistra, la schermata “scorre” verso sinistra; da destra entrano dei “rami” generati a caso, e il cuculo deve schivarli.

La finestra è occupata da una griglia suddivisa virtualmente in righe e colonne.

Il cuculo è posizionato sulla prima colonna della griglia, e può essere mosso verso l’alto o verso il basso con le frecce della tastiera.

Un timeout periodico fa muovere di una casella verso sinistra tutti i rami già sulla griglia, e genera sulla prima colonna a destra dei nuovi rami. Ogni ramo occupa una singola cella, per semplicità.

Bisogna muovere il cuculo su e giù in modo che schivi i rami che gli arrivano da destra.

Se il cuculo muovendosi si scontra con un ramo, muore. Se sta fermo e da destra gli arriva un ramo, muore.

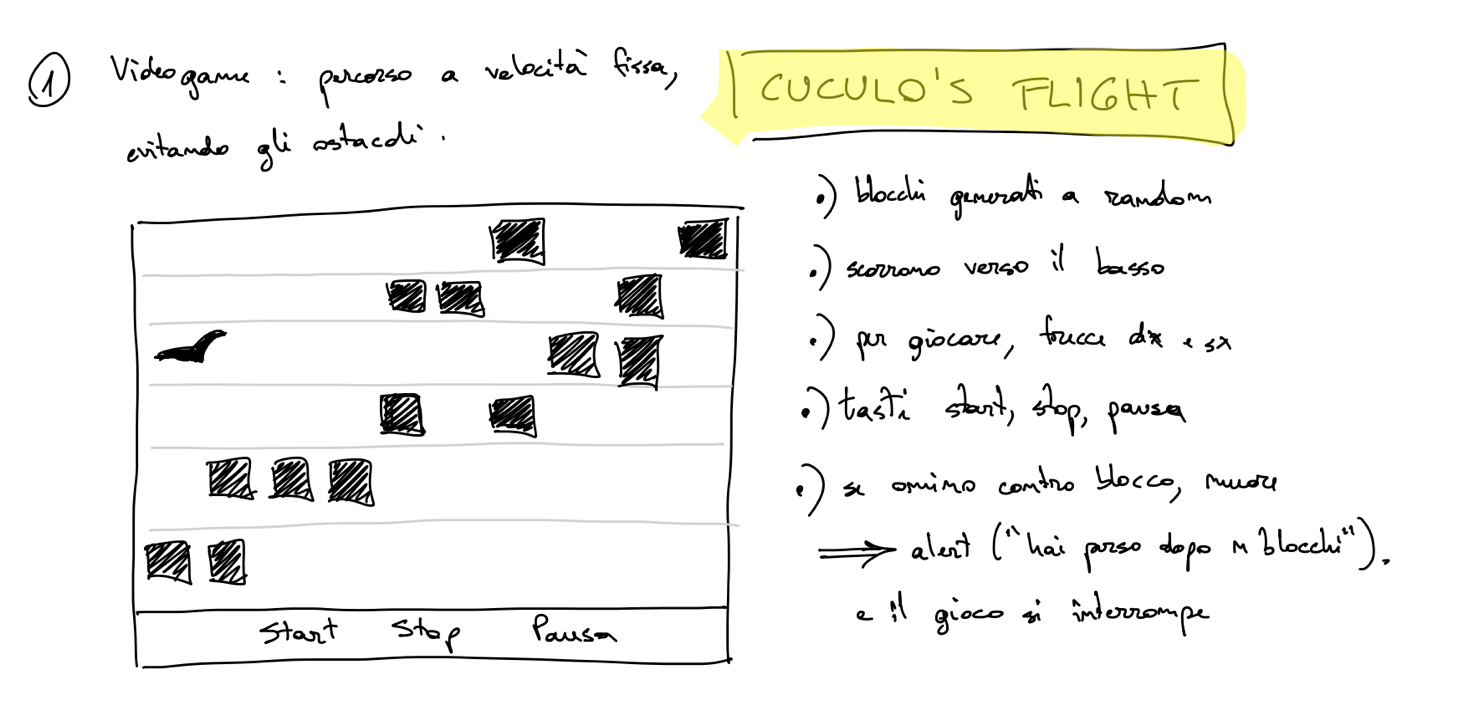
Ad ogni avanzamento dei rami, un contatore viene incrementato.

Quando il cuculo muore, deve comparire un alert che dice quant’è il contatore (cioè quanto lontano ha volato il cuculo prima di schiantarsi).

Il cuculo e i rami vanno realizzati con delle immagini opportune (i rami possono anche avere tutti la stessa immagine).

Due tasti:

* Start: il gioco parte da zero (o da dove si era interrotto)
* Stop: il gioco va in pausa



# Zipped photos

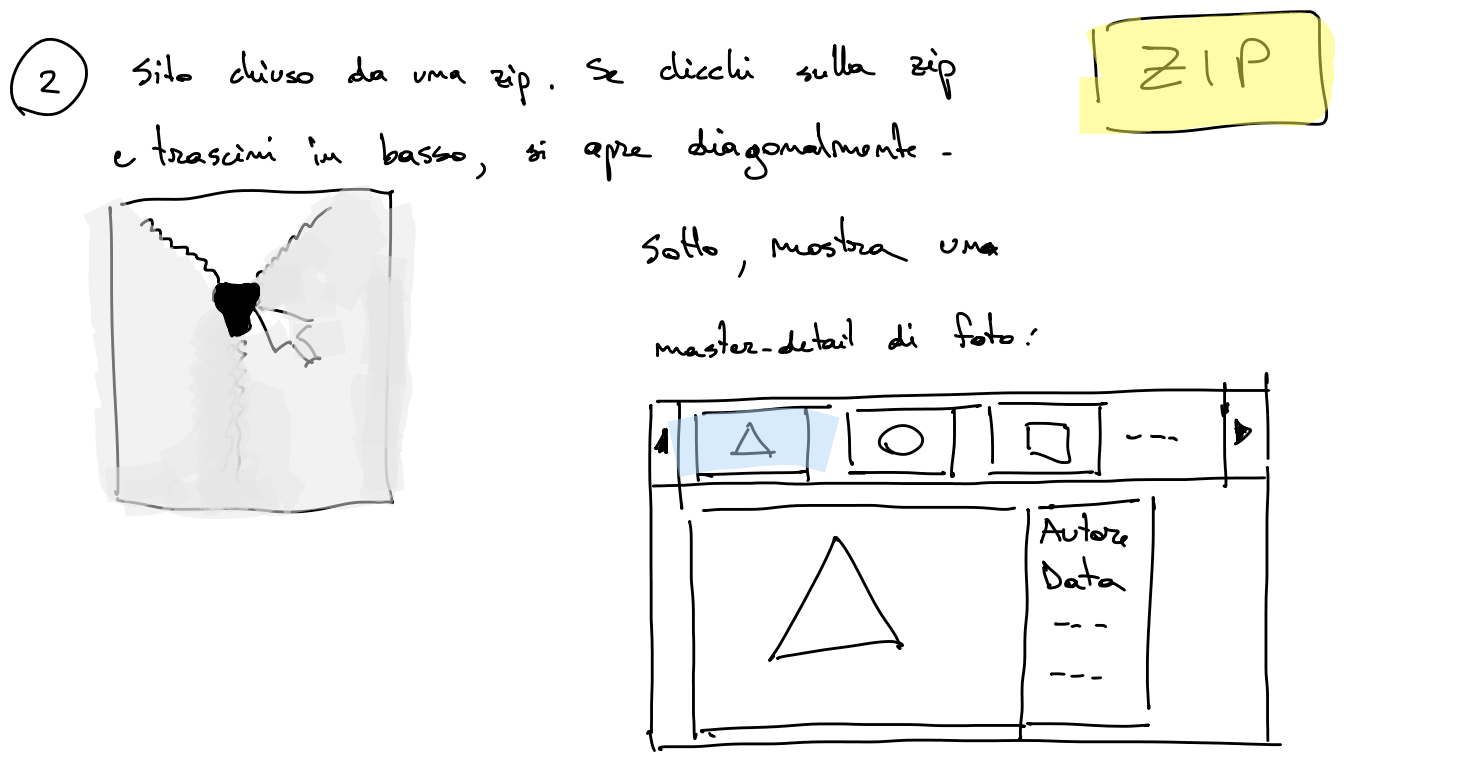
Inizialmente la pagina mostra una zip verticale chiusa, allineata al centro.

In alto c’è il cursore della zip. Trascinandolo verso il basso con il mouse, la zip si apre, e le due corsie si allargano a 45°.

“Dietro” la zip aperta si deve vedere il contenuto effettivo della pagina, che è uno slideshow di foto.

Una parte superiore mostra le miniature delle foto.

Quando clicco su una miniatura, nella parte grande centrale viene mostrata la foto ingrandita. La foto a lato ha un pannellino con le sue meta-informazioni (autore, data, dimensioni, grandezza del file…).



# Domestic zoo

Devo gestire una pensione per animali domestici, la quale ha 4 box in cui poter allocare animali.

Ogni animale può essere allocato in un box a giornate intere (ad esempio: metto il gatto Bobby nel box 1 nelle giornate di lunedì 1 marzo e martedì 2 marzo).

Il sito web ha due sezioni: una griglia box-date, e l’anagrafica di animali.

All’inizio le due sezioni compaiono come due Expander collassati.

La prima sezione mostra una griglia dove le righe sono i vari box, e le colonne sono i vari giorni dell’anno. I giorni dello stesso mese sono raggruppati, e un “sovra-header” indica il mese.

Posso scrollare verso destra o verso sinistra per vedere i vari mesi del calendario.

Quando clicco su una cella della griglia, mi si apre un dialogo che mi permette di inserire un nuovo animale. Questo dialogo mi mostra una DropDownList con l’elenco di animali, e due tasti “OK” e “Annulla”. Nella DropDownList devono essere presenti solo gli animali che posso effettivamente inserire in quel giorno in quel box. Ad esempio, se un animale è già stato inserito, non sarà presente.

La capacità di un singolo box è la seguente: posso allocarci fino a un massimo di 3 gatti, oppure al massimo 1 cane.

Ad esempio, se un box ha già 1-2 gatti dentro, posso metterci solo un terzo gatto, e quindi dovrò togliere dalla DropDownList tutti i cani.

Quando sul dialogo premo OK, sulla cella compare una striscia colorata con il nome dell’animale inserito. Ovviamente nel caso dei cani la striscia occuperà tutta l’altezza della cella, mentre per i gatti occuperà un terzo dell’altezza.

Ogni animale deve avere un colore diverso.

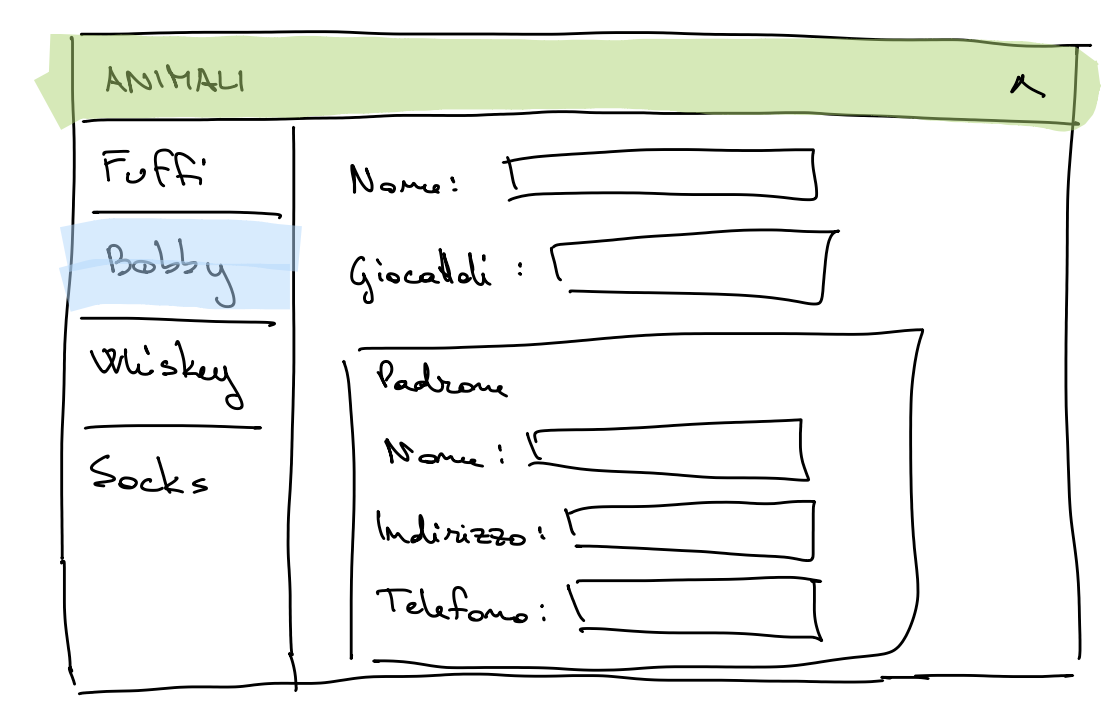
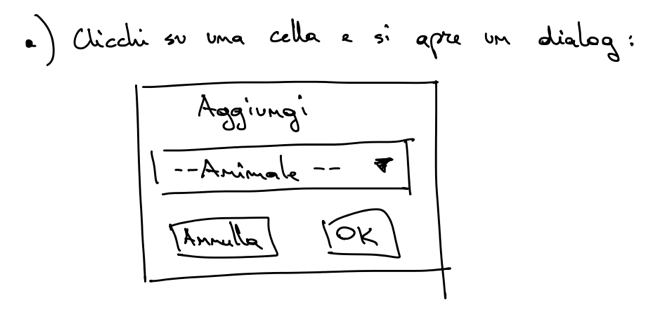
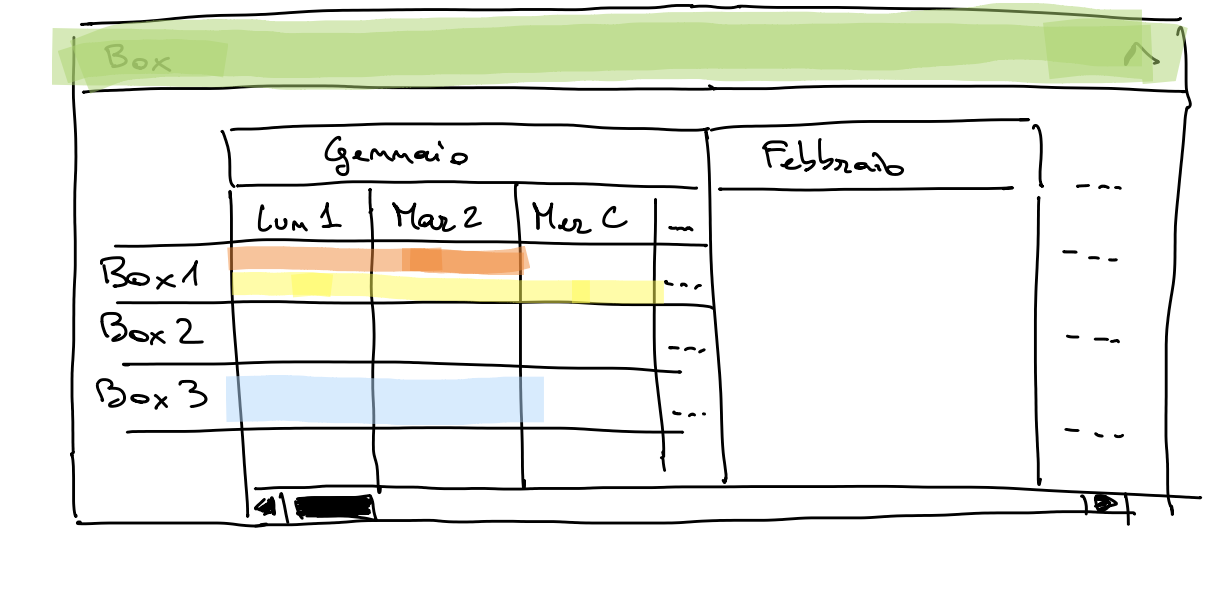
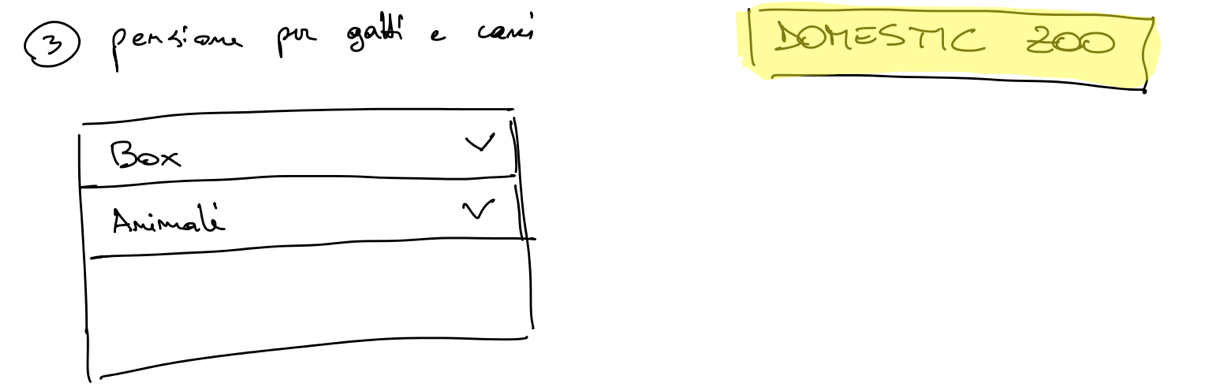
Quando clicco su una striscia colorata, un alert mi chiede conferma: “vuoi togliere questo animale dal box?”, e se l’utente preme sì, l’animale non è più allocato in quel box in quella giornata.

La seconda sezione è di tipo master-detail.

Una sezione a sinistra mostra l’elenco di animali. Cliccando su un animale, la parte di detail mostra tutte le informazioni specifiche di quell’animale, e cioè:

* Nome, Giocattoli usati, data di nascita dell’animale;
* Un riquadro specifico con le informazioni del suo padrone (Nome, Indirizzo, Telefono)

L’anagrafica degli animali va creata interamente con JavaScript dietro le quinte. NON è necessario avere un tasto “aggiungi animale” con i campi compilabili.



# Business is business

Ho un catalogo di prodotti, ognuno con nome, prezzo/unità.

Sono divisi in diversi tipi:

* Prodotti a volume: hanno il prezzo/unità che è espresso in €/m3
* Prodotti a superficie: hanno il prezzo/unità che è espresso in €/m2
* Prodotti a unità: hanno il prezzo/unità che è espresso in €/pezzo

Il catalogo è già pronto, non servono schermate di aggiunta prodotti.

L’utente deve poter creare un preventivo per i suoi clienti. Un preventivo ha una lista di uno o più prodotti, con il totale in fondo.

In cima al preventivo, un tasto “aggiungi prodotto” apre un dialogo.

Nel dialogo all’inizio è visibile solo una DropDownList con l’elenco di prodotti del catalogo.

Quando l’utente seleziona un prodotto, allora viene mostrata la seconda parte del dialogo, in cui deve inserire i parametri. Ad esempio, se ha selezionato un prodotto a volume, ci saranno 3 input per altezza, larghezza e profondità. Se ha selezionato un prodotto a unità, ci sarà un unico input per inserire quante unità di prodotto si vogliono.

Oltre ai parametri specifici del tipo di prodotto, poi c’è sempre un input per indicare lo sconto da applicare.

Oltre agli input modificabili, ci sono degli altri input readonly che mostrano i valori calcolati:

* Quantità totale (ad esempio se ho un prodotto a superficie, sarà base \* altezza);
* Prezzo/unità (preso dal prodotto del catalogo selezionato)
* Prezzo totale (quantità totale \* prezzo/unità)
* Prezzo finale (prezzo totale con l’applicazione dello sconto)

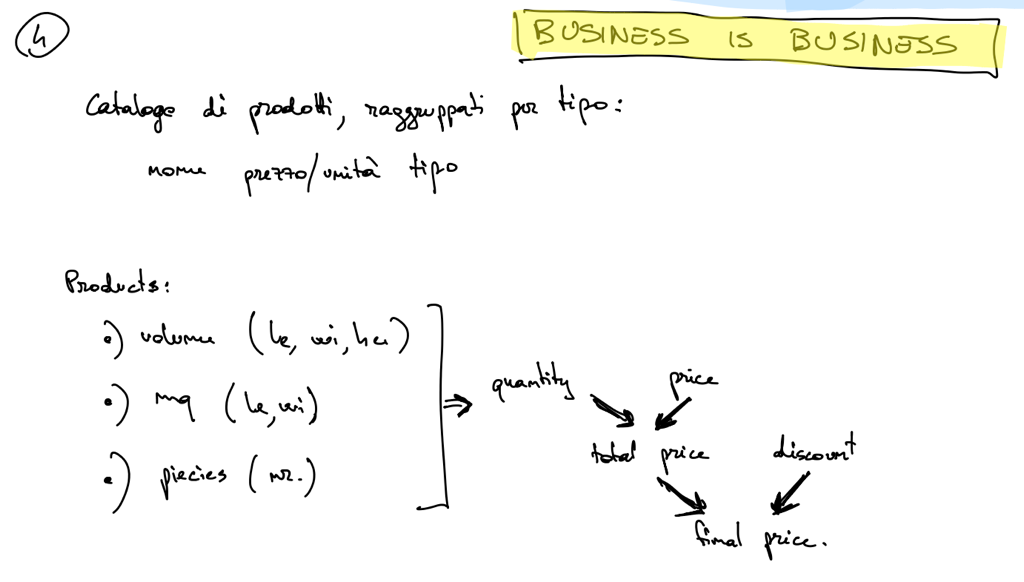
In fondo al dialogo ci sono i pulsanti “OK” e “Annulla”.

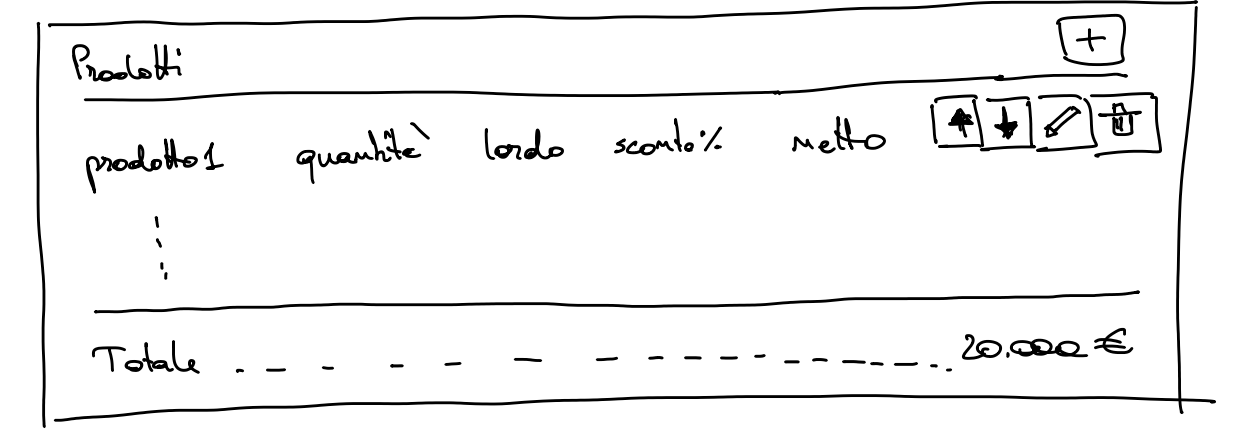
Se l’utente fa “OK”, il prodotto appena creato viene aggiunto in fondo al preventivo.

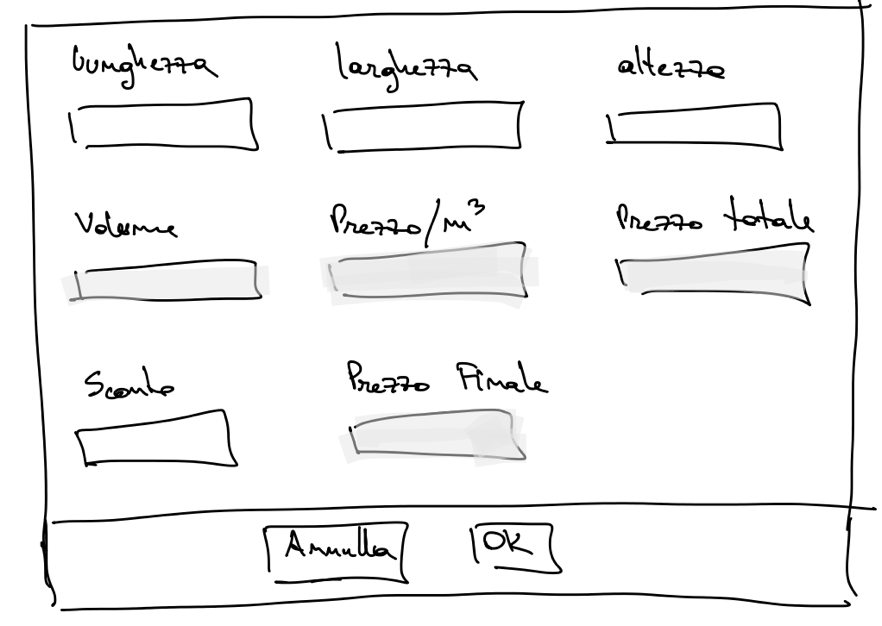
Ogni prodotto mostrato in elenco ha sulla destra 4 pulsanti:

* “su” e “giù” per spostare il prodotto su e giù nella lista;
* “edit”: si apre lo stesso dialogo di cui sopra, ma con le informazioni del prodotto selezionato. L’utente può quindi modificare i valori e salvare premendo “OK”
* “delete”: il prodotto viene rimosso dal preventivo.

Una riga in fondo mostra il totale del preventivo, calcolato facendo la somma del prezzo finale di tutti i prodotti aggiunti.







# Super Mario Busy

I personaggi di Super Mario vogliono organizzarsi con un calendario condiviso.

La pagina web è di tipo master-detail.

A sinistra, il master mostra l’elenco di personaggi (Mario, Luigi, Toad…). Ogni personaggio è mostrato in una riga che ha una piccola foto/immagine e il nome.

Nella parte detail, viene mostrato un calendario in cui ogni colonna è una giornata, suddivisa verticamente nelle varie ore del giorno.

Sul nome del giorno c’è un pulsante “+”, cliccando il quale si apre un dialogo che mi permette di aggiungere un evento al calendario.

Il dialogo permette di indicare nome dell’evento, ora di inizio e ora di fine, e gli altri personaggi coinvolti (selezione multipla!).

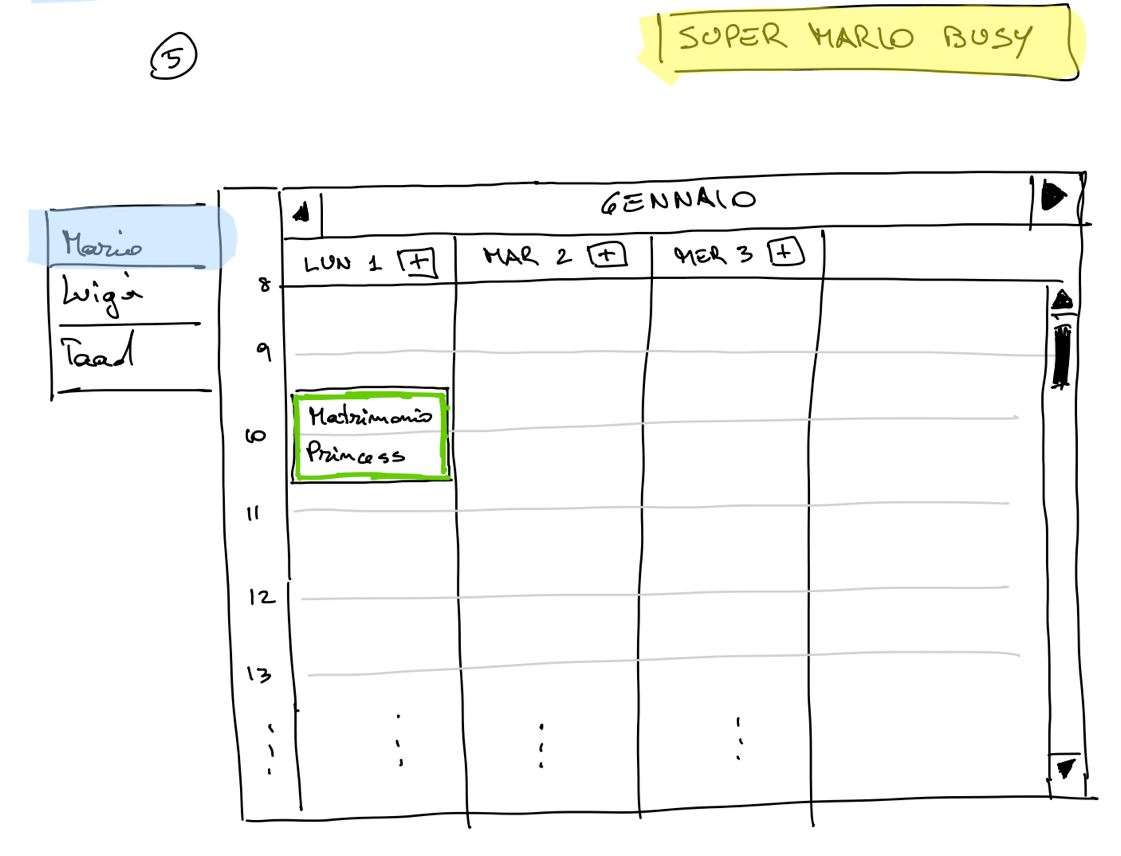
Quando l’utente clicca “OK”, bisogna controllare che l’evento NON si accavalli ad altri eventi (NON possono esserci eventi sovrapposti), e bisogna controllarlo anche per gli altri personaggi indicati.

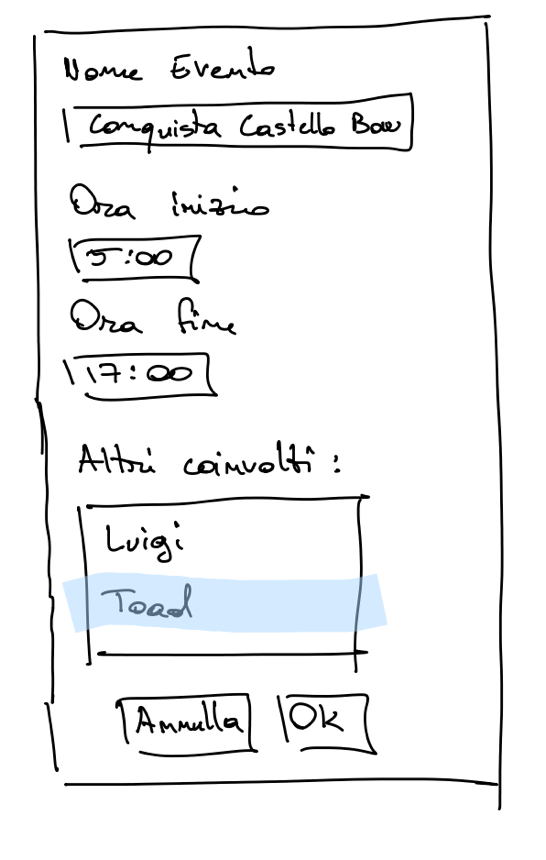
Se è tutto ok, l’evento viene inserito in calendario. In quel caso sul giorno scelto viene mostrato un rettangolo colorato che copre tutte le ore dell’evento. Ogni rettangolo mostra in cima il nome dell’evento.

Ogni evento deve avere un colore diverso, e quel colore deve essere lo stesso per tutti i personaggi coinvolti nell’evento.

Quindi, se aggiungo su Mario un evento il 25 gennaio dalle 11 alle 13 colorato in rosso che coinvolge anche Luigi, quando poi vado nel dettaglio di Luigi devo trovarmi lo stesso rettangolo rosso sullo stesso giorno e stesse ore.

Per modificare un evento, clicco sul rettangolo colorato. Mi si riapre il dialogo, con i campi già popolati.





# Calculator Pro

Realizzare una calcolatrice per calcoli complessi.

C’è un controllo di “input” in cui l’utente può inserire:

* Le 10 cifre
* Il punto decimale .
* Le operazioni + - \* / % ^ (cioè somma, sottrazione, moltiplicazione, divisione, modulo, potenza)
* Le parentesi tonde ( )

Quando l’utente preme Invio, viene analizzata la stringa con i calcoli, e viene calcolato il risultato.

Nel riquadro dello storico dei calcoli viene inserita una nuova riga con due celle: a sinistra l’espressione appena inserita, e a destra il risultato calcolato.

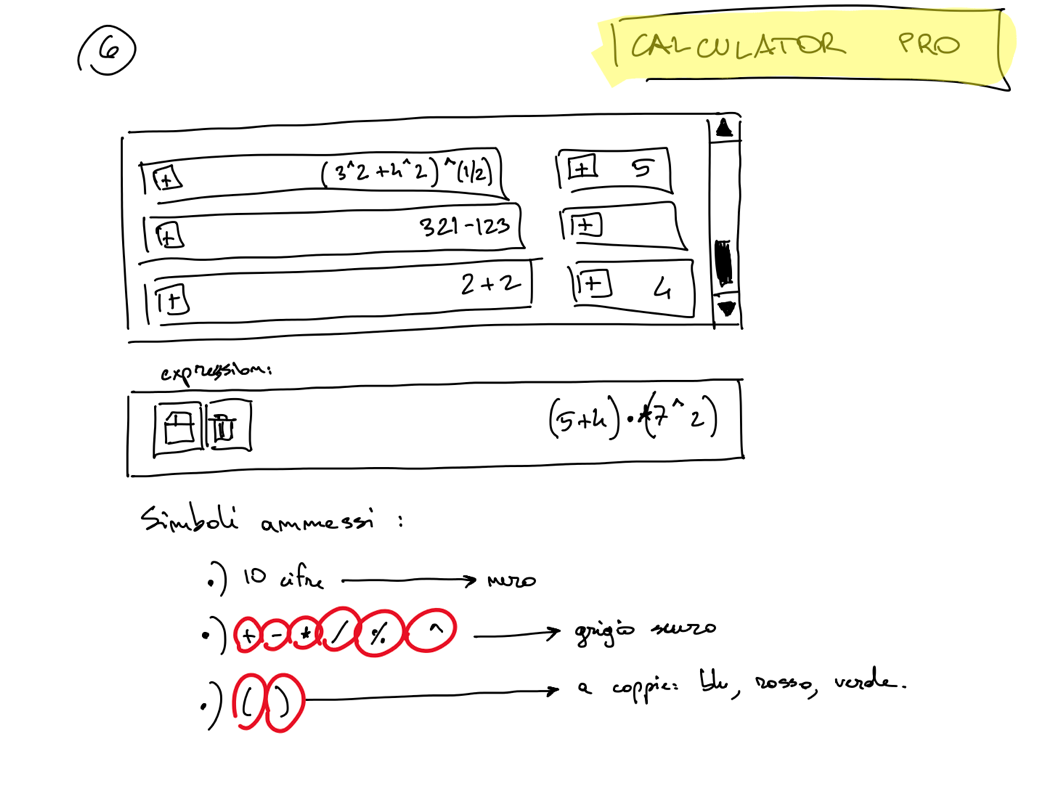
Se la stringa inserita non è corretta (ad esempio manca una parentesi), non viene inserita nello storico; invece un alert avvisa l’utente che c’è qualche errore.

L’espressione inserita deve colorare il testo in base al tipo di caratteri:

* Le cifre sono in nero
* Gli operatori sono in blu
* Le parentesi sono in rosso

Nello storico, in ogni espressione e in ogni risultato c’è un tasto “+”. Se premuto, aggiunge il contenuto della cella in coda ai caratteri inseriti dall’utente. Ad esempio, se l’utente ha scritto 7 \*, e clicca il “+” sulla cella dello storico con scritto dentro 120 - 85, l’espressione nell’input principale diventa: 7 \* 120 - 85.

La sezione con lo storico può essere messa sopra o sotto l’input principale, basta che abbia un’altezza massima: se l’utente fra troppi calcoli, la sezione di storico diventa scrollabile.



# Dottor House Dati

Un ospedale deve gestire l’anagrafica dei suoi pazienti.

Ho una griglia con un elenco di persone.

Ogni riga è un Expander: se clicco il pulsante a destra, si espande verso il basso una sezione di dettaglio della persona.

Dentro ogni Expander ho due sezioni, una per il dettaglio della persona, e una per l’elenco di servizi sanitari usati.

Nella prima sezione vengono mostrati foto, nome, cognome, sesso, data di nascita, indirizzo della persona.

Nella seconda sezione c’è la lista di servizi utilizzati dalla persona, ogni riga è a sua volta un Expander.

Dentro l’Expander del singolo servizio, seleziono le caratteristiche.

Nella prima DropDownList, il tipo di servizio (ci sono servizi di tipo visita (ad esempio: visita generale, visita oncologica, visita pediatrica), e servizi di tipo intervento chirurgico (intervento al cuore, intervento al cervello, intervento al colon, …).

In una seconda DropDownList seleziono il medico che ha effettuato il servizio. Bisogna creare a monte un elenco di medici, e assegnare ad ogni medico un certo servizio (ad esempio Mario Rossi e Luigi Neri sono medici per le visite oncologiche; Anna Gialli, Giorgio Verdi e Federico Bruni fanno interventi al cuore, ecc). Quando seleziono un servizio per una persona, poi nella DropDownList dei medici devo vedere solo i medici che fanno quel servizio.

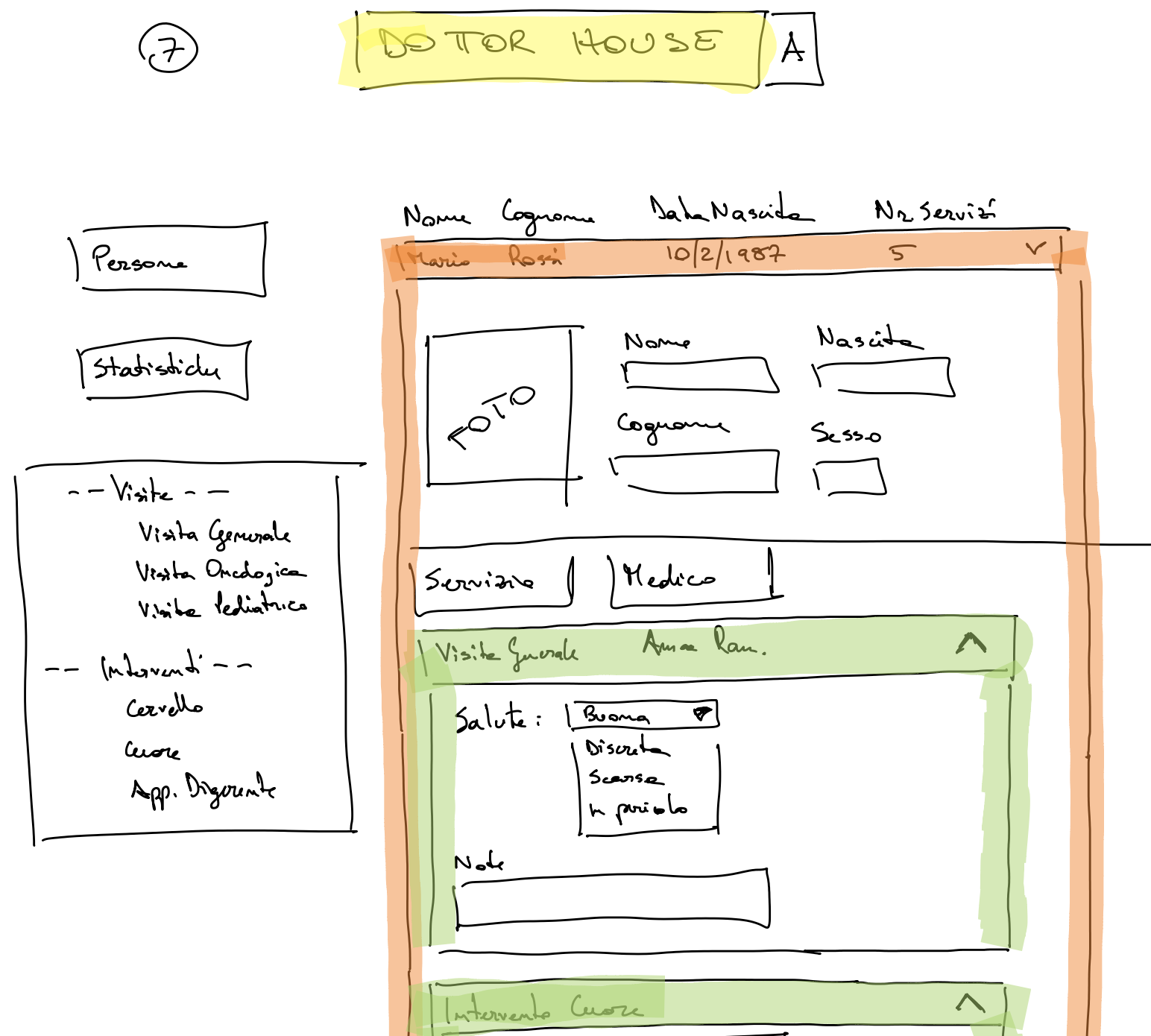
In base al tipo di servizio, le informazioni da visualizzare sono diverse:

* Se è una visita, devo mostrare:
  + Una DropDownList da cui indicare la salute del paziente Buona Salute, Qualche Acciacco, Salute Cagionevole, Pericolo Di Vita;
  + Un date-picker da cui selezionare la data della visita;
  + Un campo Note, multilinea.
* Se è un intervento chirurgico:
  + Una DropDownList da cui indicare il risultato dell’intervento: Successo, Qualche Complicazione, Decesso;
  + Un elenco di input in cui inserire quanti by-pass, valvole, punti di sutura, placche metalliche sono stati usati nell’intervento;
  + Un date-picker da cui selezionare la data dell’intervento
  + Un campo note multilinea.

Degli opportuni tasti permettono di:

* Aggiungere una persona;
* Aggiungere un servizio dentro il dettaglio di una persona.

Tutti i campi sono editabili quando apro i vari Expander.



# Dottor House Graphics

Creare un’anagrafica pronta di persone e servizi sanitari, come per l’esercizio precedente (cioè c’è un elenco di persone, ognuna con alcune proprietà e un elenco di servizi di cui ha usufruito, e in ogni servizio le varie caratteristiche…)

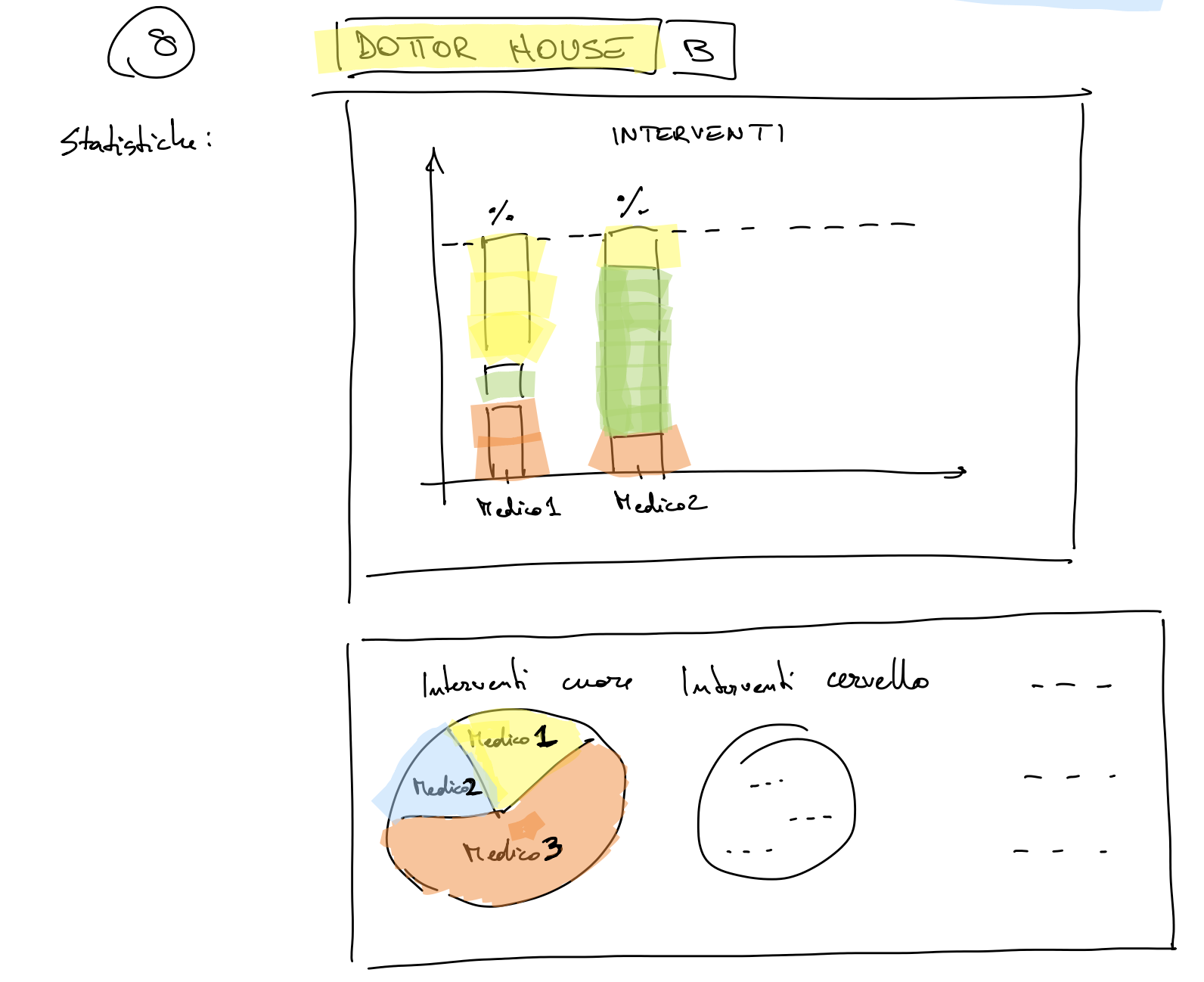
In questo esercizio, vanno realizzati dei grafici di report dei dati.

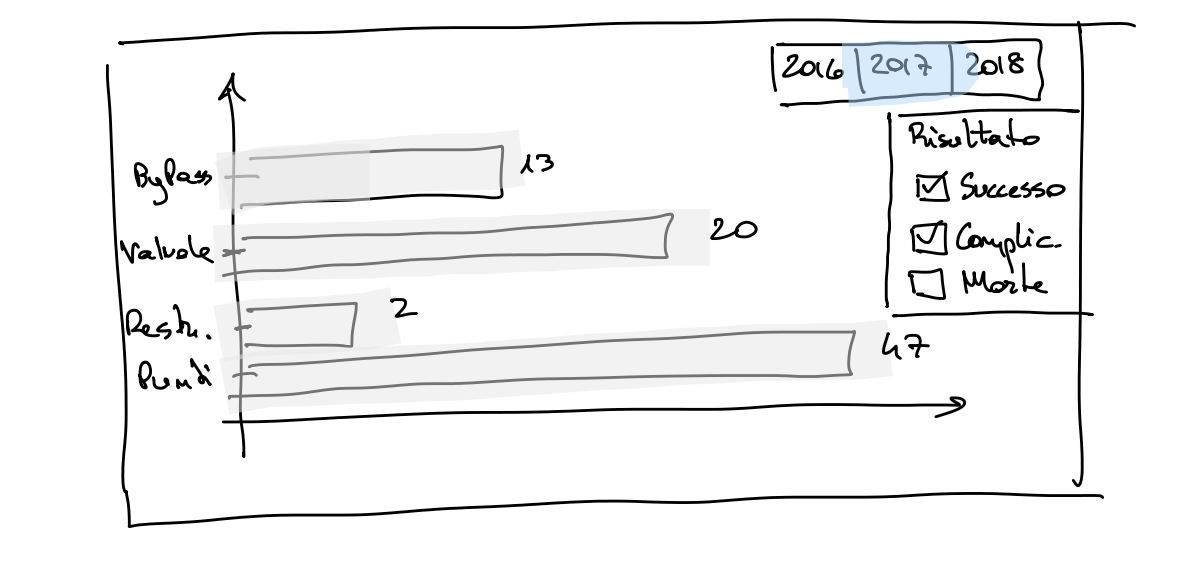
**Percentuale successi**. Un diagramma a barre verticali, ogni barra è un medico, ed è divisa in tre fasce, colorate in verde, giallo, rosso. Le fasce rappresentano gli interventi che quel medico ha fatto con successo, complicazioni, o decesso, in percentuale (quindi le barre dei vari medici sono tutte alte uguali in totale).

**Operatività medici.** Un diagramma a torta per ogni tipo di servizio; le fette della torta sono i medici che hanno effettuato i servizi; ad esempio se il medico A ha effettuato 20 visite generali, il medico B 10 e il medico C 10, avremo la torta divisa in quattro: 50% + 25% + 25%.

**Statistiche utilizzo materiale**. Un diagramma a barre orizzontali. Ogni barra è un tipo di materiale usato negli interventi chirurgici (by-pass, valvole, punti…). Ogni barra è lunga in proporzione a quante unità sono state usate. A destra del grafico, ho un filtro per anno (quindi vedo le quantità solo sul singolo anno), e un altro filtro mi permette di selezionare il risultato dell’intervento (successo, complicazioni, decesso). In base all’anno scelto e alle check-box del risultato, il grafico si aggiorna.

Perché le statistiche siano serie, va generata una quantità importante di dati (ad esempio, almeno 7-8 tipi di servizi, almeno 30 medici, almeno 50 persone, …).





# Comics Chess

Realizzare una scacchiera, in cui ogni pezzo corrisponde ad un personaggio famoso (che sia reale o dei cartoni animati). Modificare un po’ le immagini in modo che sia chiaro se appartiene ai bianchi o ai neri.

Quando l’utente clicca su un pezzo della scacchiera, quello si evidenzia.

L’utente poi seleziona la casella di destinazione. I casi sono:

* La casella non è un valido movimento per il pezzo (o perché il pezzo non può fare quella mossa, o perché è già occupata da un pezzo amico), quindi un alert avvisa dell’errore;
* La casella è libera: il pezzo viene animato per spostarsi verso quella casella; el caso del cavallo, il movimento deve essere ad angolo (cioè fatto di due passaggi)
* La casella è occupata da un pezzo nemico: il movimento viene fatto, e il pezzo nemico viene rimosso dalla scacchiera e posizionato sopra o sotto la scacchiera tra i pezzi mangiati.

Il gioco è per due giocatori, quindi a turno posso muovere un bianco, un nero, un bianco…

