Hardware and software Helper – Design documentation

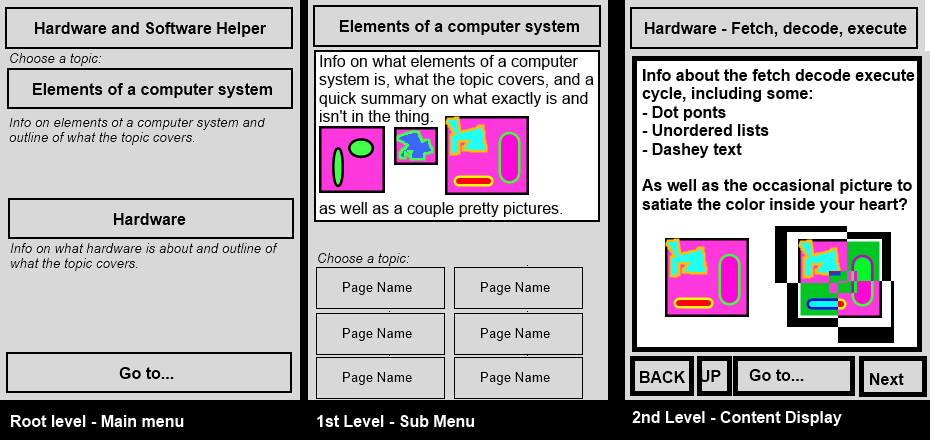
# Product specifications

After consulting with the client to obtain information about the desired product with B. Carpenter (the assignment sheet), we have concluded that the following information needs to be included in the product:

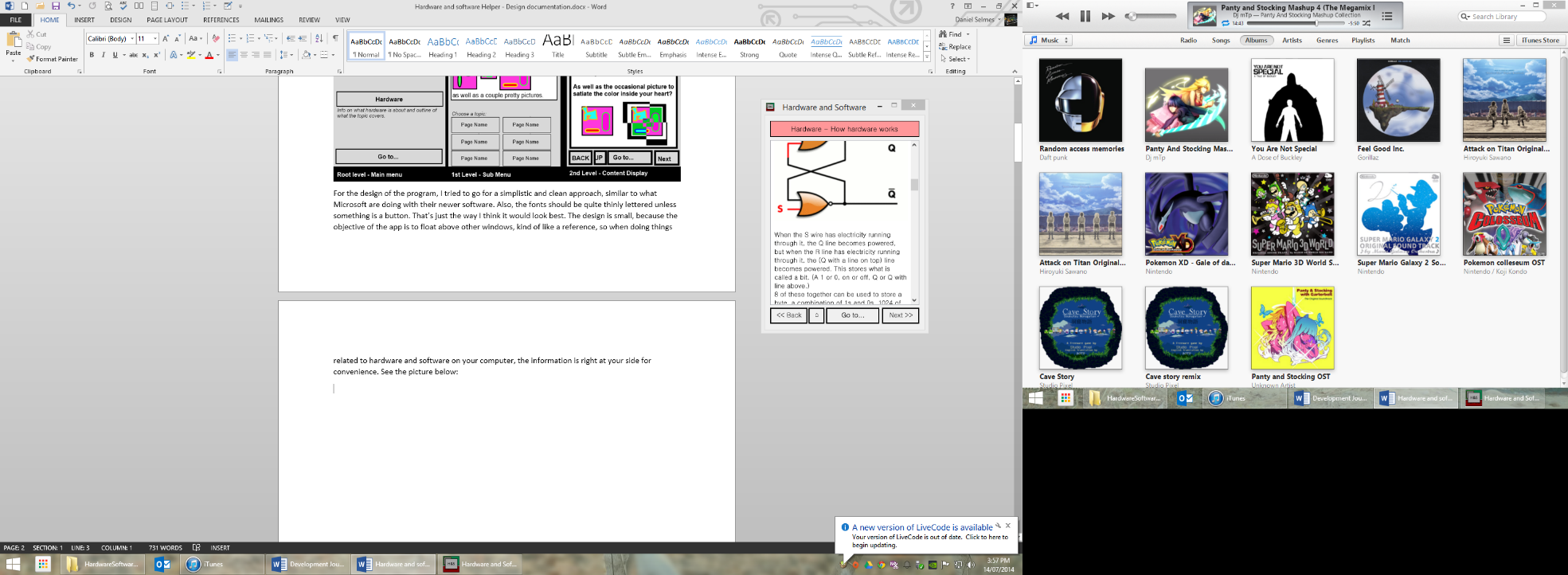
* Elements of a computer system:
  + Hardware
  + Software
  + Personnel
  + Procedures
  + Data
* Hardware
  + The functions of Input, output, process, storage and control
  + How components such as CPUs, Storage devices, and input and output devices achieve their purpose.
  + Trends and developments in computer hardware
  + Processing of software instructions by hardware; the Fetch-Execute cycle
  + The existence of minimum hardware requirements to run software

The product should be neat looking, easy and intuitive to use, and contain multiple types of media including text, images and video.

# Screen design

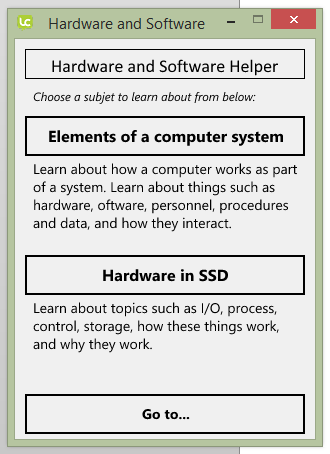


For the design of the program, I tried to go for a simplistic and clean approach, similar to what Microsoft are doing with their newer software. Also, the fonts should be quite thinly lettered unless something is a button. That’s just the way I think it would look best. The design is small, because the objective of the app is to float above other windows, kind of like a reference, so when doing things related to hardware and software on your computer, the information is right at your side for convenience. See the picture below:

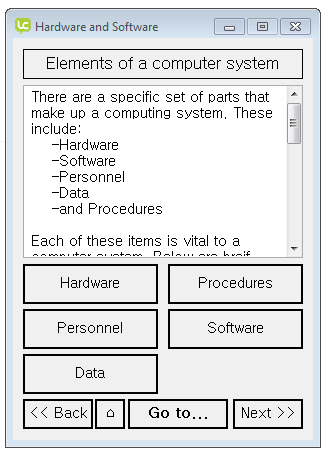


Here, the application is sitting on screen beside the user’s document, so they can take notes from it without having to tab out between it and their current document.

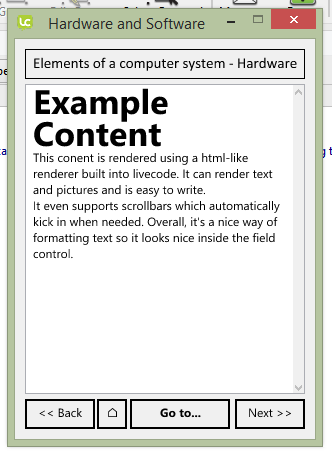
# Storyboard



This is the start-up screen, and the main menu. It is the first screen the user will see, and is designed to guide them to the section of the application they wish to go to. There is also a drop-down to jump to any scren, for users to easily find what they are looking for in only 2 clicks.

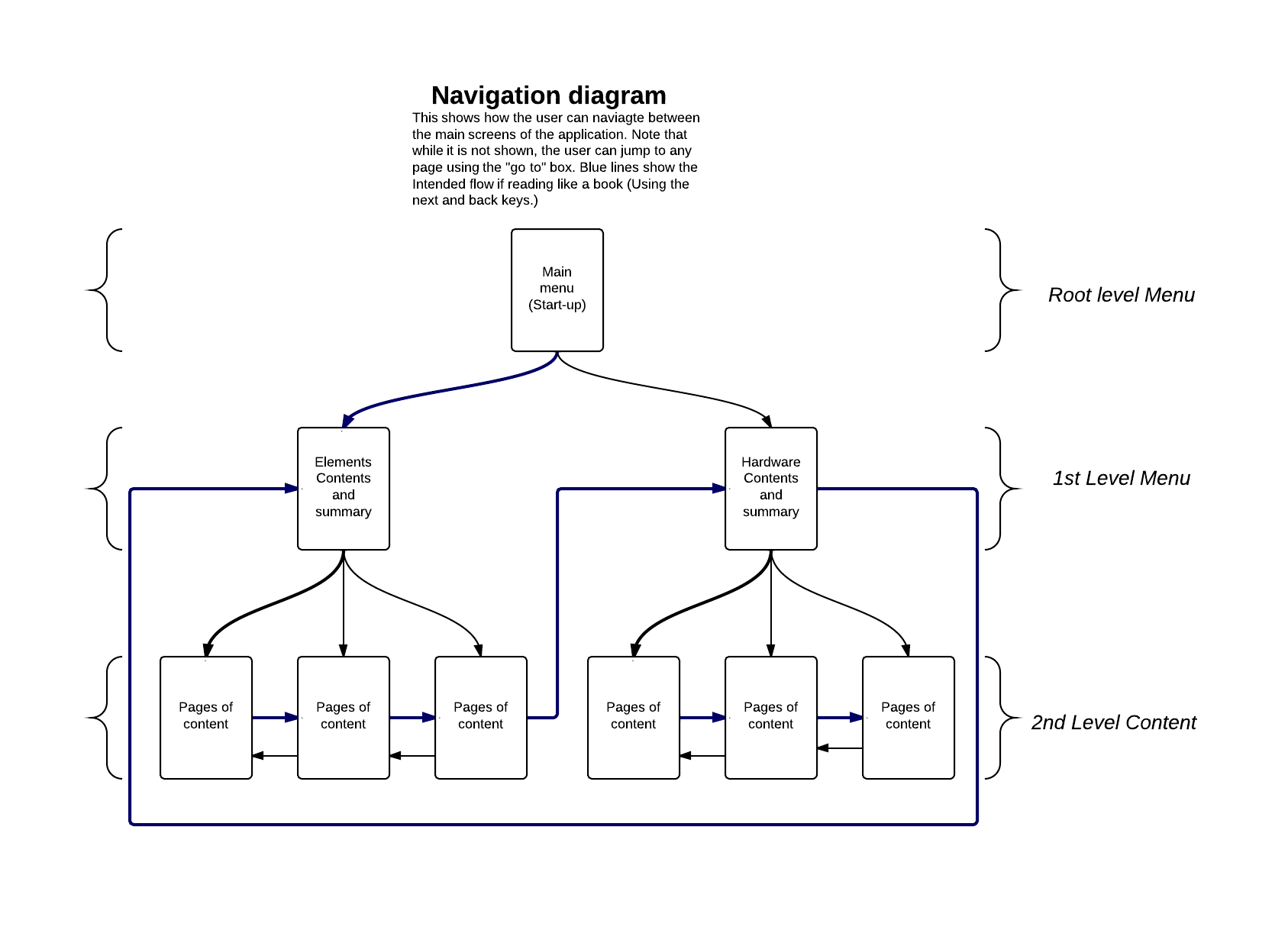


This is the module menu. It shows the user some basic info about the module that the user is viewing, and has buttons to jump to each of the pages in the module. It also has the familiar go to, back, next and home buttons for quickly navigating to where the user wants to go. It’s also designed with a similar layout to the content pages to help the user feel at home with the interface as soon as possible.



This screen is an example of a content page with just some text, but pictures can easily be implemented in the future. The menu pages eventually lead to this page, see the navigation hierarchy for more info.

# Navigation



This diagram shows how the screens of the application relate to each other. The top level screen appears when the program starts, and from there, the user can either filter down the menus to the content or jump straight to the content. I opted for the tree based approach as it works well when presenting categorised information.

# Design of text, images and whitespace, colour and layout

I opted for the layouts I used because they fill up most of the screen with content but allow titles and controls in a non-intrusive looking way. The layouts are consistent, clean and neat for easy navigation and helping the user understand where they are in the application, as well as being aesthetically pleasing. I also made most of the images as wide as the content box so that they would line up with the edges of the text AND be easily visible. The colour in the title areas helps users to know where they are in the app and what to expect when browsing. The fonts choses were sans serif as it’s proven that sans-serif is easier to read and display on a screen, as well as the fonts themselves looking clean.

# Error handling

There is basic error handling now available instructing users how to report a bug and offering an email address to email the bug to. I’ve found from testing that bugs don’t really ever happen, but if one decides to come up, we should be able to fix that quickly.

# Code

The code for this project is very simple, just setting up an animation and then changing card. It’s still coded meaningfully though, with the screen objects having meaningful names. The code is also pretty readable, and comments aren’t really necessary thanks to LiveCode’s syntax.

# Support

User support is provided easily on the product website so users can be familiar on how to download, install and operate the application. There is also context sensitive help.