Computer Science: A-level Project

[redacted]

CANDIDATE NUMBER[redacted]

CENTRE NUMBER: [redacted]

[redacted] - Candidate Number: 4099 - Centre number: 16802

PC Builder

Contents

Analysis

Problem Definition

A personal computer (PC) can defined as a multipurpose computer that used to run a variety of programs. Although this can include all kinds of technology such as Phones, tablets and games consoles, the 'PC' term is Commonly in reference to a desktop computer, which is a computer used at a desk with peripherals such as a monitor, keyboard and mouse plugged into it to provide an interface to interact with the desktop computer programs. These computer programs can range from browsers (for World Wide Web access), file managers (to store, move and delete files), office work (creating reports, presentation etc.). The main reason why me and many other 'PC gamers' would get a desktop computer nowadays is for playing video games, but some of the newer video games are increasingly straining of the PC's hardware and may not be able to run on the PC's hardware. That is where PC building and modifying, which make use of the upgradability of a PC, has become the main means to get the required hardware to run these hardware intensive programs, especially video games.

PC building is the collection and assembling of the components used in a PC. This process test many skills such as budget management, to get the best parts for your budget and make the most of your money. It also test technical knowledge, as a lot of research into the compatibility of the parts you are collecting, before you waste money on parts that do not work together. Once you have the part you then have to assemble it, which too take skills such as cable management, a delicate touch and will need curtain tools. However, if you do build a PC it can save lots of money compared to if you get someone else to build your PC or buy a whole, new one. It also means that you can upgrade your PC incrementally as parts get older and bottleneck the system. Another benefit of this is that there are less discarded comparts because as old components can be re-sold and re-used to get money back instead of thrown away resulting in environmental damage.

The public tend not to build a PC themselves because it is seen as risky, complicated and time consuming. Even some tech savvy people are put off building a PC, as finding and comparing the best parts for their build can be time consuming and difficult; especially when they have to commit purchase once it is made.

I plan to create an environment in which both newbies and experienced PC builders can mess around with a large variety of components to find which configuration works best for them before going on to make a purchase. However, trough decomposition and other computational methods, this problem can be broken down in to smaller solvable components, which can be resolved with existing technology.

Computational methods can be used to develop a solution

Decomposition will allow me to breakdown the problem into smaller problems (or modules), which I can solve independently. By resolving items such as, Interface, storage management, data structures and theirs constituent decomposition elements, it will allow to slowly build on the product an can independently resolve these problems to resolve and build the product. Another benefit of using decomposition is that I can get my product testers and stakeholders to trial and provide recommendations on smaller already completed modules before I integrate in into my code. The modularity of using decomposition means that, debugging and making changes to the program code and functionality is a lot more practical than, the more complicated overhaul that would be expected from a liner or 'hardwired' program that would reduce the efficiency of the program and be more time constraining.

A major part of the problem is how the storage of the statistics and information about the PC parts is handled. I can used databases, which can be easily navigated to find information about parts the user is interested in, using database queries. This also allows us to connect the all users to the same information by using the internet to link user's devices to the same database, which on a server that can be managed though a DBMS. As a result, the latest information such as prices and new products can be updated on one database and will appear on multiple user device as they are all connected to the same database rather than a local one. A non- computational solution could be to use a catalogue but it is expensive to produce large number to provide access many user. It is also had to update it with new information because it physically printed on catalogue books that are miles apart, so the best way is to produce new and up to date ones, which take time, more money and would leave out of date catalogues still available in the outside world. The central database is very inexpensive compared to a catalogue. It easy to update and can practically store larger amounts of information in a smaller space yet easy accessible space.

Pattern recognition would be useful in providing recommended parts to the users. This can be done by logging at the components that tend to be used together and using that to inform other users of combinations that have worked before and may work for them. This helps to create a more welcoming user experience and to help the companies making the components to know what the demand like for their products.

I will have to use abstraction in my project especially in the representation of the PC's components. For example, I will not be able to run a fully accurate simulation of what the PC being built would be like from the hardware of the end user, as a result I would use a form of abstraction in performance modelling, which would the statistics and components' data to replicate the expected performance of the PC. A benefit of using a computational solution is that, on a computer mathematical calculations needed for this modelling can be performed on the device. This can include calculations, such as calculating the total price of the components, the user wants or comparing performance figures of a wide variety of PC parts. This can be done on the user's devices processor, rather than the user spending a long time calculating the total, which could lead to miscalculations though human error. It is also a lot quicker than and more user friendly if the computer sorts the PC parts rather than the user. Although this I more achievable way of getting the performance figures of the PC being built, it is not 100% accurate as there are some factors that cat be accounted for in the statistics.

Another way I can implement abstraction is that I plan to have a simple interface as to avoid congesting the user with information. Because of this, I will have to disregard and reconstruct elements of building a PC when implementing it to the 2D interface provided by an app on a phone or a web page. That could include elements like cable management, static proofing or the probability of component failure. Having these features degrade the user experience and is not essential in the act of testing and experimenting with parts in the program. Lastly, in terms of graphical abstraction, I do not need to go into the exact detail of how each component looks in real life. Instead, a I could use a picture of the component or an animated sprite with the key feature of that component can be used instead as it not only reduces the graphical processing and file size, but it can convey enough information for the user to know what component they are using.

I will be using algorithmic thinking in the implementation of the backend code. I will use algorithmic thinking to make decisions on what instructions need to execute. This allows me to the code I will write clearly though flowcharts or pseudocode. I can also use algorithmic thinking to increase line efficiency by recalling repeated lines of instructions though functions and procedures.

Initial thoughts on developing a solution

In order to produce the product, I will first do some research on both existing products and possible stakeholders who have an interest in the product or would like to help in the production of it. I will also carry out interviews with and make a questionnaire for possible end users and stake holders, this is to collect information on what the end user's preference would like for the product. After which, I would compile an initial set of requirements and a success criteria to follow. From there I can begin the design phases by first decomposing the problem and planning on what data structures, algorithms and user interface I will use. When developing (implementing) the code I want to use the Agile Project Management method, which means I would implement my designs into the product in small sections aided by my decomposition. For each of these sections I will interactively and independently, develop, test and evaluate before I implement it into my main product. As I reach landmarks I will evaluate the project as a whole to measure its effectiveness in meeting the success criteria and to allow stakeholders to see to development is proceeding.

The main areas in which I see complexity coming from would be in the database management and managing of the interface. User friendly is a top priority in terms of meaning the success of this product. A wide variety of information and PC components in the database would help that and so would a welcoming interface for the user to get use to PC building

I can see this problem mainly being solved by a mobile app. This is because apps are easy to access for the end users and can be distributed though application stores. Another way to produce this product is through a web site. This would mean that both phones and existing PC or laptops should be able to access it though the world wide web. I can possibly produce both an app and an embedded program in a website at the same time by using an engine.

Stakeholders

The Stakeholders I am looking for this project are people with PC building experience to see, if my solution provides any benefit to their previous experiences. I would also like people who are new to PC building to see whether they would be more inclined to Build or find it easier to build a PC using the app. Finally, my final group of stakeholders are those from a web or app development company, who can both provide me with advice for how to create an app and can provide industry supported feedback on what I create.

My first stakeholder is Alexander Ford is an example of my projected end user. Mr Ford does not have much experience for building PCs. He has an old PC and is looking to upgrade or possibly perchance a new one. His feedback would be useful in understanding how people with little experience cope with the product. I will keep in close contact with Mr Ford especially through testing of the user interface and user experience. I accept that his feedback is from a single user's perspective and other people may have different feedback but he can give me an indication of whether it is satisfying the end users' needs. With a working solution, Mr Ford would more easily be able to find and compare parts to help him to build his own PC or at least order the parts.

The next stakeholder is from Edwards Office computer repair shop. They are a small shop but have decades worth of experience in building, assembling and repairing computers. With them as a stakeholder, they can help me validate how much of an improving the product has over more traditional methods pf pc building. They can also give me advice on what PC parts that I should include in my solution, and the accuracy of the product to the user. They may use my finished product as an app to enable their customers to select PC parts for the people at Edwards Office to assemble or fix at the shop, this in return will invite more customers to their services. A possible side feature of the product would be a helpline run by Edward's Office to help those struggling to build a p or going through hardware issues with their PC.

WebX Solutions are a software development firm. They have years of experience in developing apps and web browser software. This is useful to me as they can give advice on what works and what does not in terms of designing a user-friendly user experience and user interface. They also have facilities like network servers that can host databases. In addition to that, they also have a variety of hardware for me to test my prototypes on to check for compatibility on different systems, which is a key step in assuring quality for the end user.
For a technology, based company like Webx Solutions there is a need to frequently, upgrade their computer systems. An app or software like the one I am planning will allow for the CTO or whoever's job it is to execute the systems upgrade, to more easily refine the new PC systems that they want to implement before they make a purchase.

Stakeholder [contact]	Stake in the project(why are they interested)	How they can help me	Project roll	Expectations from project and me	influence	Risk if they are not involved	Strategy to keep the stake holder involved

Alex Ford [email: altford@yahoo.co.uk]	Mr Ford is planning to build a PC but has very little experience of doing so. This fits my target demographic.	Mr Ford is a possible end user, and would help by testing the application to see if I helps hip to build a PC	Product tester (quality control)	To produce a functioning prototype for him to test and provide feed back	mid	Low – I can find more people with little to no PC building experience	Provide Mr Ford with frequent updates and an eastern egg in the application
Edwards Office [email: edwardsofficerepairs@gmail.com]	A working application would provide an easier way for them to get orders from customers looking to repair or assemble their PC	They can advise on the best PC parts out there for the application , they can also be used as a customer's helpline to support people building their PC	PC building advisor and Customer help service	They expect me to reference their business as to create free advertisement for them. They also expect to see regular updates to make sure that there is still potential for the application	low	Low- although tree advice would be nice not having them working alongside will not stop me from making prototypes a proof of concept that an application like this works in the real world.	I will provide them with regular updates and if a share of any revenue made if the product develops far enough to be published. This is along with their advertisement.
WebX Solutions ltd [phone: +44 20 8500 1303 Or the email contact form on their website: webxsolution.com/contact.html]	Webx are relatively small; company with less than 10 project development staff. If I make a good product there is potential for them to hire me or take me on an apprentice to work with them.	They have years of experience in creating apps and websites. They can provide me with both facilities and advice to how to create the application	Providing facilities (servers and a range of testing hardware to test the application)	Give them functioning prototypes that show good development. In need to make sure that these prototypes are developed on schedule as to not waste their time. By doing this they can see that	high	High – If they do not get involved, I will struggle to find test servers to host the application . In addition, most importantly, I will to gain access to their range of devices to help me test the	I will provide them with scheduled updates on the product's progression to show that it still has momentum, I will also offer shared development licences if they like the product and want

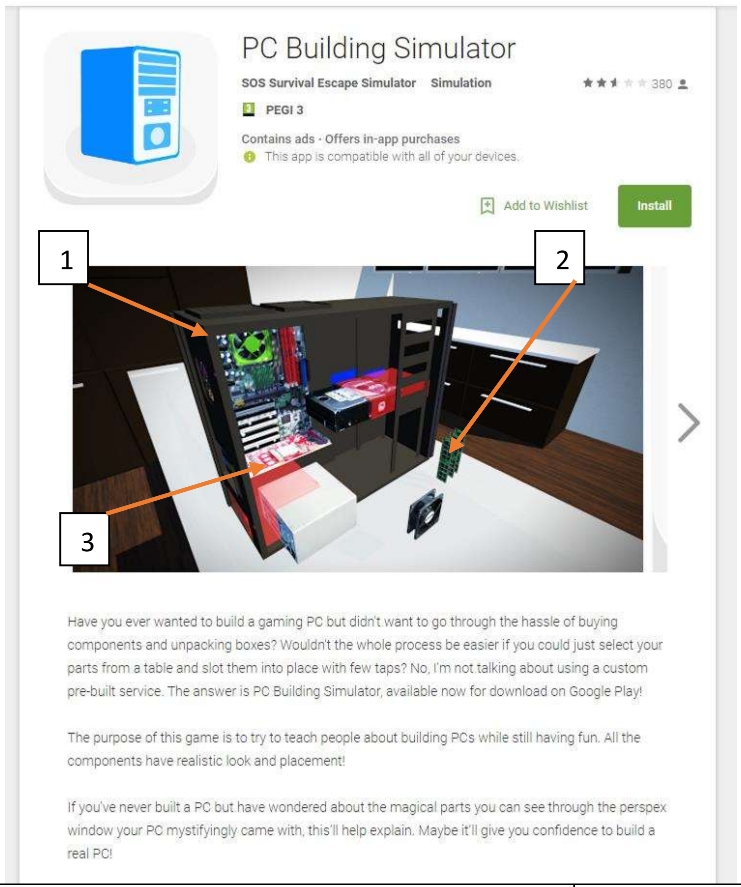
	Since I am young, this gives the chance to raise a model employee suited to their company.			my product is on track and still has potential.		product on different platforms.	to develop in further they can put it in their résumé.

Research

Existing solutions

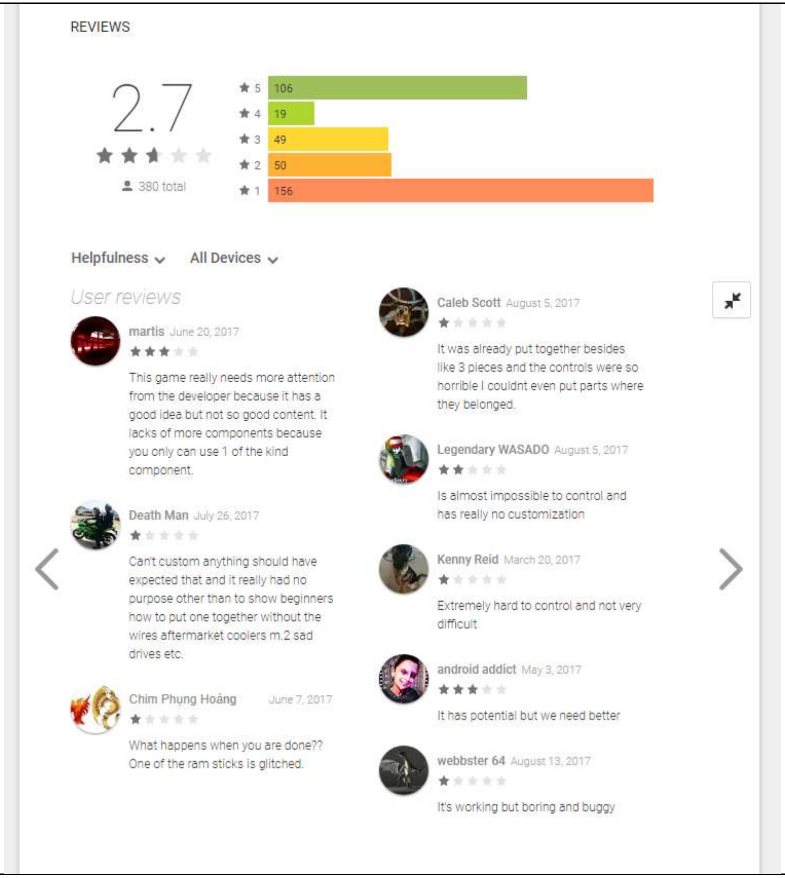
I did some investigation into any current apps or software that is used for building PC, especially on mobile devices to see, firstly if there is a well-known solution out there and if there were any problems in existing solutions that I can solve with mine.

PC building simulator

 PC Building Simulator SOS Survival Escape Simulator Simulation 380 PEGI 3 Contains ads · Offers in-app purchases This app is compatible with all of your devices. Have you ever wanted to build a gaming PC but didn't want to go through the hassle of buying components and unpacking boxes? Wouldn't the whole process be easier if you could just select your parts from a table and slot them into place with few taps? No, I'm not talking about using a custom pre-built service. The answer is PC Building Simulator, available now for download on Google Play! The purpose of this game is to try to teach people about building PCs while still having fun. All the components have realistic look and placement! If you've never built a PC but have wondered about the magical parts you can see through the perspex window your PC mystifyingly came with, this'll help explain. Maybe it'll give you confidence to build a real PC!	Name: PC Building Simulator By: SOS Survival Escape Simulator Dowanloads:10,000(aprox.) Rating: 2.7/5(380 reveiews) PC building allows you build a gaming PC without the hassle of buying components and unpacking boxes. it was made to teach people about buildidng PCs and it does a good job with its 3D graphics. **Annotations** (1)Shows that all the components are rendered in 3D, which enables the user to move around and look at what they are building from all angles. (2)The app uses drag and drop controls to easy move parts into and out the PC. (3)The segment where the selected part goes is highlighted this is a useful feature for a user who has no PC building experience.
Comments & reviews	

Most of the comments say it is a good idea and has good potential. However, it is much unpolished with plenty of bugs and glitches. Some comments also mentioned that the objects were difficult to control and move in a 3D space. A possible solution to this could be to use a more abstracted 2D interface. Finally, some comments mentioned a lack in verity of PC parts to choose from which ruined the user experience as the game soon become repetitive.

I also tried this app myself and found the same problems in my own personal experience of using their app.



REVIEWS

2.7

380 total

★ 5	106
★ 4	19
★ 3	49
★ 2	50
★ 1	156

Helpfulness All Devices

User reviews

martis June 20, 2017 ★★★
This game really needs more attention from the developer because it has a good idea but not so good content. It lacks of more components because you only can use 1 of the kind component.

Death Man July 26, 2017 ★
Can't custom anything should have expected that and it really had no purpose other than to show beginners how to put one together without the wires aftermarket coolers m.2 sad drives etc.

Chim Phụng Hoàng June 7, 2017 ★
What happens when you are done?? One of the ram sticks is glitched.

Caleb Scott August 5, 2017 ★
It was already put together besides like 3 pieces and the controls were so horrible I couldnt even put parts where they belonged.

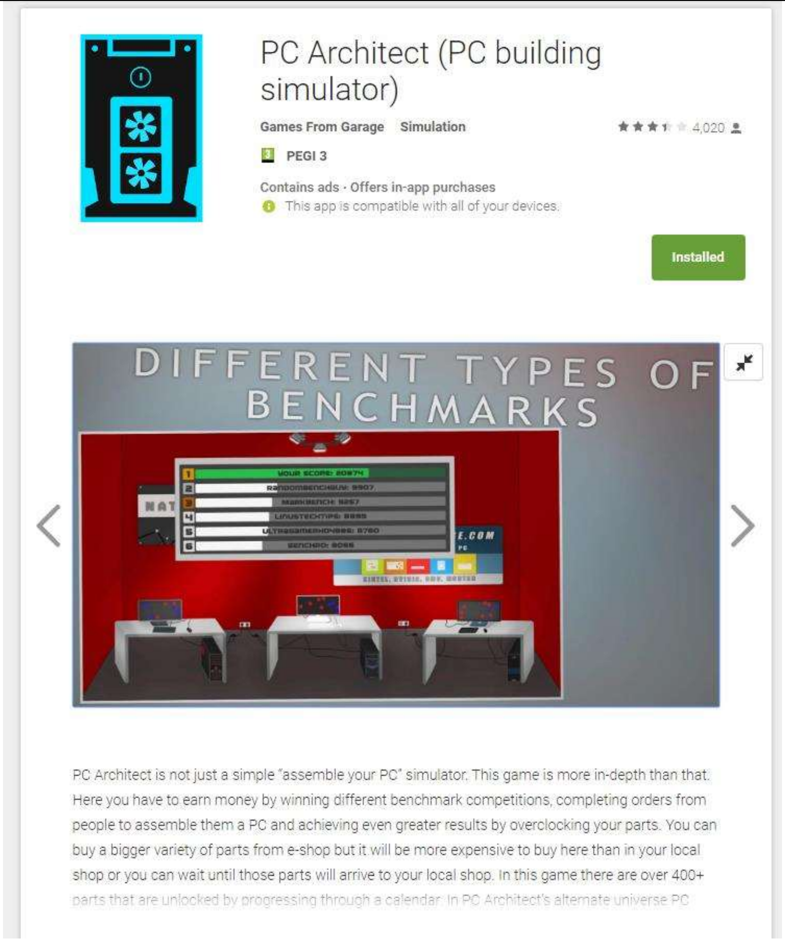
Legendary WASADO August 5, 2017 ★★
Is almost impossible to control and has really no customization

Kenny Reid March 20, 2017 ★
Extremely hard to control and not very difficult

android addict May 3, 2017 ★★★
It has potential but we need better

webbster 64 August 13, 2017 ★
It's working but boring and buggy

PC Architect (PC building simulator)



PC Architect (PC building simulator)

Games From Garage Simulation 4,020

PEGI 3

Contains ads · Offers in-app purchases

This app is compatible with all of your devices.

Installed

PC Architect is not just a simple "assemble your PC" simulator. This game is more in-depth than that. Here you have to earn money by winning different benchmark competitions, completing orders from people to assemble them a PC and achieving even greater results by overclocking your parts. You can buy a bigger variety of parts from e-shop but it will be more expensive to buy here than in your local shop or you can wait until those parts will arrive to your local shop. In this game there are over 400+ parts that are unlocked by progressing through a calendar. In PC Architect's alternate universe PC

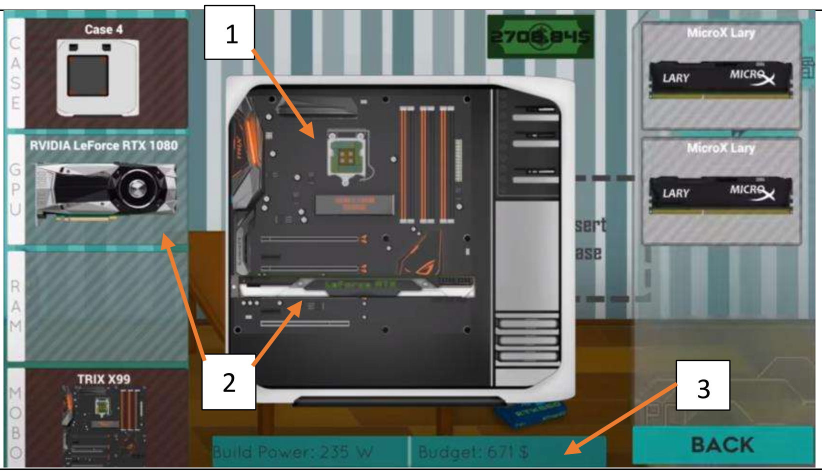
Name: PC Architect (PC Building Simulator)
By: Games From Garage
Dowanloads:100,000(aprox.)
Rating: 3.5/5(4,021 reveiews)

From what I saw this the best pc building related app/game on the google play store. It is story driven game about the player starting a building company form nothing. It features over 400 different computer parts, which are unbranded or feature fake brand names that reference real world alternatives. This is because of licensing complications, which is something I need to watch out for in my own application. The game feature many elements from the pc building community including: benchmark tests (as seen by the screenshot), shopping for parts in shops and online, unboxing and assembly of the computers. These are features I see as useful in my solution. Another feature is chance of failure and when assembling the computer. Although it Is a real life part of building computers, I see this as more of a plot device for the story line the game depicts and would not be as useful in my application.

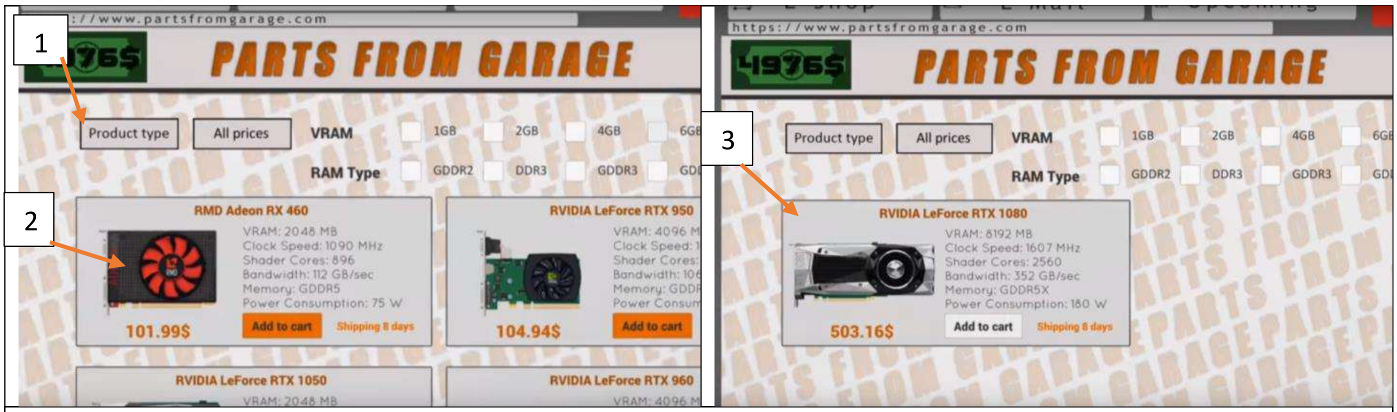
There are two main traits that I see would be common between my product and this game.
In my app I would like the user to get a visual impression of how to assemble the parts when once they have ordered and delivered form the app. PC Architect has done a good job of this with its building screen. Another feature that I can also see being used in my app is the process of searching for parts. In the game this is done by the "e-shop", which allow for different search methods and categories to sort the 400+ parts in the game so the user can find what they want. I see this a useful in my own product as well. I have done more detailed annotations of these screens

Annotations (Building screen)

here is a screenshot of the PC architect building screen. I like the format in which the building information is displayed in this screen.
(1) This is a 2D representation of inside the case. It provides a good representation of the slots on the motherboard and the how much space there is inside the case to parts into.
(2) Due to the 2D perspective being from overhead onto the motherboard it is



impossible to what the components look like from a sideways point of view. The developers overcome this by displaying the side on perspective of the parts in a scroll menu on the left side where the parts are selected from.	3)This is a status bar which give information about the items currently in the virtual PC. In the screenshot it shows the power consumption of the parts current in the virtual PC and also how much spending money remains in the virtual PC's budget.



RMD Adeon RX 460
VRAM: 2048 MB
Clock Speed: 1090 MHz
Shader Cores: 896
Bandwidth: 112 GB/sec
Memory: GDDR5
Power Consumption: 75 W
101.99$

RVIDIA LeForce RTX 950
VRAM: 4096 M
104.94$

RVIDIA LeForce RTX 1050
VRAM: 2048 MB

RVIDIA LeForce RTX 960
VRAM: 4096 M

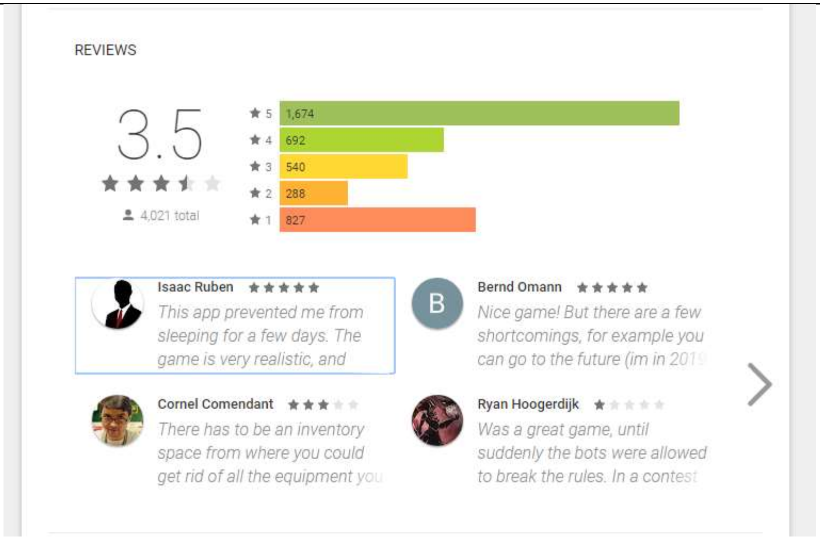
RVIDIA LeForce RTX 1080
VRAM: 8192 MB
Clock Speed: 1607 MHz
Shader Cores: 2560
Bandwidth: 352 GB/sec
Memory: GDDR5X
Power Consumption: 180 W
503.16$

Annotations (Parts searching)

The parts searching screen is important for the user experience to be quick and efficient when looking for parts.
(1) At the top is a filter bar where the user can select product type, prices and various properties depending on the item. This means that the user no longer has to scroll through unwanted items to get to the group of items that match what they're looking for.
(2) Each item in the list has a box containing details of each product. This means that the user can clearly see the names, prise and statistics of each product. From a coding perspective it means that a procedure can be made and called on when displaying an item, no matter how may item there are. This is a very efficient way to do this.
(3) Once the filters are applied it hides all of the other product that the user is not interested in.

Comments & reviews

The majority of the recent comments were positive and the ones that deducted ratings did do because of bugs. One of the bugs that I experience was that sometimes the menu buttons were unresponsive when pressed. This is just a responsiveness issue, which can be corrected with a patch. Apart from that, the comments praise it for its variety of parts and concept of PC building. One thing it is lacking in terms of PC parts is a CPU cooler. A features that I do like is the in game customer email section, which gives you budget challenges to complete a PC build to a set of requirements. I see this a fun game mechanic that helps with the long-term use of the game.


REVIEWS

3.5
4,021 total

★ 5	1,674
★ 4	692
★ 3	540
★ 2	288
★ 1	827

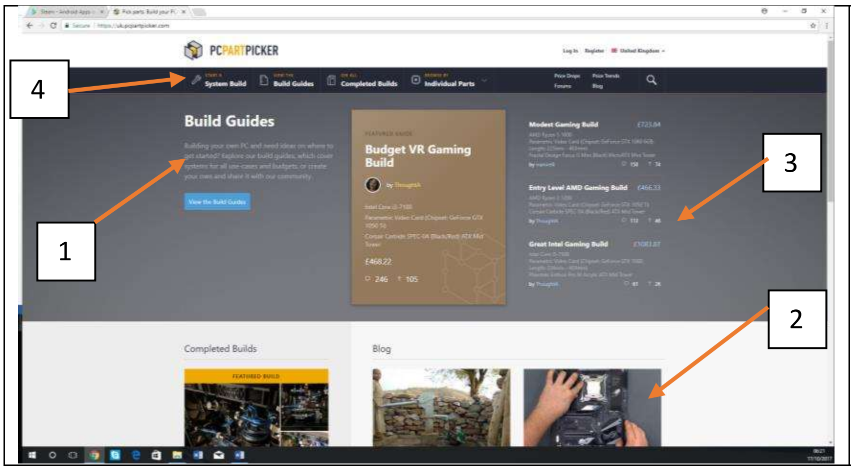
Isaac Ruben ★★★★★
This app prevented me from sleeping for a few days. The game is very realistic, and

Bernd Omann ★★★★★
Nice game! But there are a few shortcomings, for example you can go to the future (im in 2019

Cornel Comendant ★★★
There has to be an inventory space from where you could get rid of all the equipment you

Ryan Hoogerdijk ★
Was a great game, until suddenly the bots were allowed to break the rules. In a contest

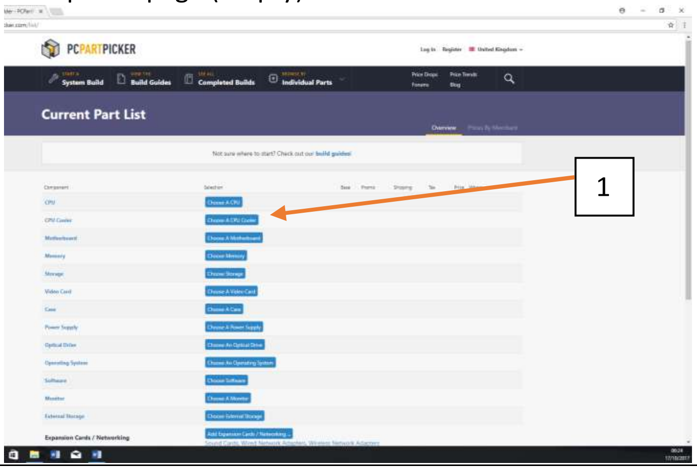
pcpartpicker.com



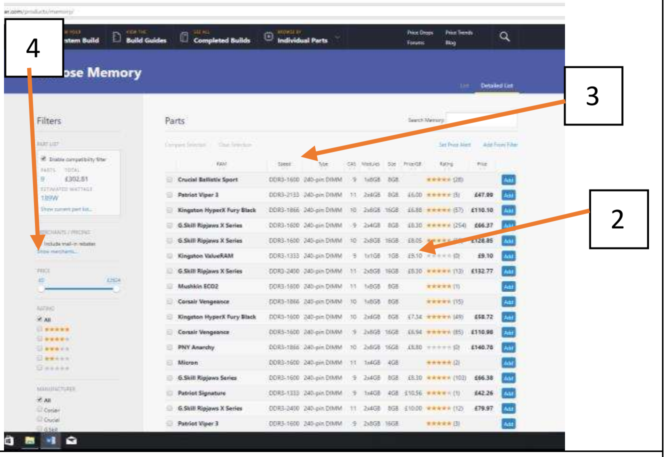
One of the biggest pc building tools is the PC part picker website. It contains PC building guides (1), blogs (2), online shops (3) and its flagship feature the part picker page (4).

It is this part picker page that apples to my project the most, I have gone into detail of how the page works.

Part picker page (empty)

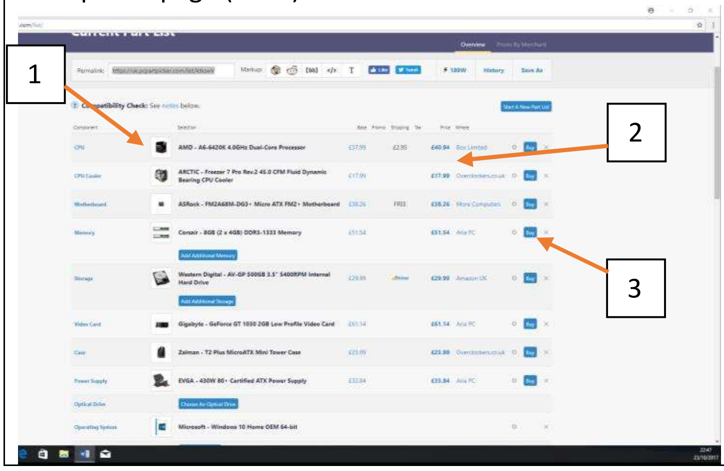


Components list



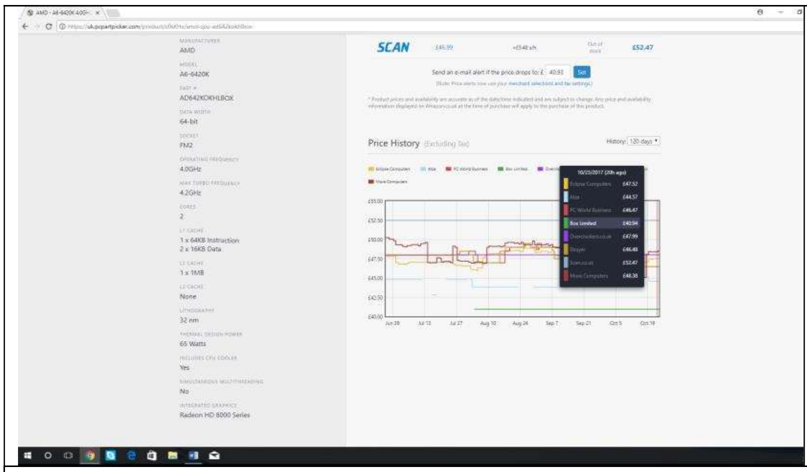
To test their page, I decided to make my own entry-level, gaming PC parts list. I set myself a £300 budget and selected parts using the add component buttons (1). This took me to the detailed list of registered parts that fit the component description ((2) the example above is for RAM). I could sort the parts by category (3), in the memory section you can sort by name, speed, socket type, price, rating and more. Also on the right of the page is a filter section (4), here is where you can filter out parts by price, rating, size and more. Then you can add these parts to your list.

Part picker page (filled)



This is what the page looks like when filled, it displays a picture of the part (1), price of part (2), any extra costs like shipping and the shop or company that distributes the parts (3). Finally, there is a button on the side of each component to buy them individually.

A nice feature is that a 'permalink', which is a custom URL that is generated with your PC parts list for you to copy and share the components of the PC. This is the one for my list. https://uk.pcpartpicker.com/list/ktkswV


AMD - A6-6420K 4.0GHz

MANUFACTURER
AMD

MODEL
A6-6420K

PART #
AD642KOKHLBOX

DATA WIDTH
64-bit

SOCKET
FM2

OPERATING FREQUENCY
4.0GHz

MAX TURBO FREQUENCY
4.2GHz

CORES
2

L1 CACHE
1 x 64KB Instruction
2 x 16KB Data

L2 CACHE
1 x 1MB

L3 CACHE
None

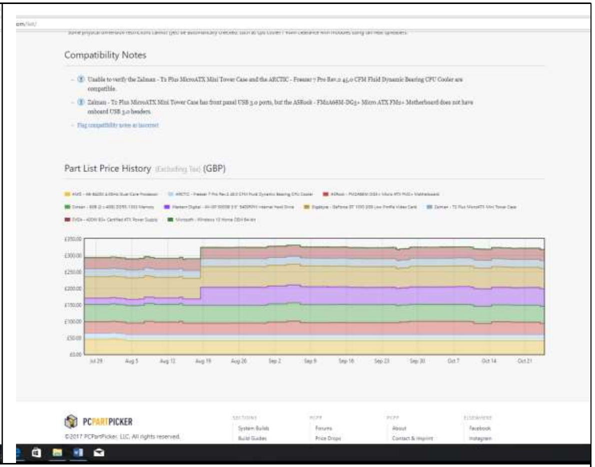
LITHOGRAPHY
32 nm

THERMAL DESIGN POWER
65 Watts

INCLUDES CPU COOLER
Yes

SIMULTANEOUS MULTITHREADING
No

INTEGRATED GRAPHICS
Radeon HD 8000 Series


Compatibility Notes

- Unable to verify the Zalman - T2 Plus MicroATX Mini Tower Case and the ARCTIC - Freezer 7 Pro Rev.2 45.0 CFM Fluid Dynamic Bearing CPU Cooler are compatible.
- Zalman - T2 Plus MicroATX Mini Tower Case has front panel USB 3.0 ports, but the ASRock - FM2A68M-DG3+ Micro ATX FM2+ Motherboard does not have onboard USB 3.0 headers.
- Flag compatibility notes as incorrect

Part List Price History (Excluding Tax) (GBP)
PCPARTPICKER
© 2017 PCPartPicker, LLC. All rights reserved.

A feature that I really like is the statistics about each part and the combined parts in the list. One of my favourite is the price history. This shows the change in price of the components from multiple shops over the last year or so. The stats above are from the price of the CPU in my list in different shops and the total price for all of the parts over time.

Conclusion

From my research and testing of existing products that relate to my problem, I have gathered that:

Polish and smoothness is key to the user's enjoyment of the product. A smooth and responsive software allows the user to quickly trial and exchange parts to see what works. An irresponsive and bug filled app was a problem and main complaint with 'PC building simulator' and 'PC Architect'. I can resolve this with my product by using my stakeholders at Webx Solutions to help test my product on different hardware to make sure the product integrates well with it.

Another feature that helps with the user experience is a wide variety of parts to choose from. 'PC building simulator' did not have any parts to choose from and resulted in a short, unfulfilling experience, which reflected in its reviews and poor rating. 'PC Architect' and 'pcpartpicker.com' have a great variety is parts to pick from this means there are more combinations of parts that reflect the real world and allows the user to trial more parts to fit their needs and budget.
The wide variety of parts also requires a filter or sorting system as seen in 'PC Architect' and 'pcpartpicker.com' to help the user find the parts that suit them quickly. An issue I may also have with my selection of parts is the licencing to use real brands and products in my product. 'PC Architect' overcomes tis with the use of fake names that hint towards real brands.

Finally, user interaction with this product is also important. I like the ease of use when searching in a list view as seen in 'PC Architect' and 'pcpartpicker.com'. I also preferred the 2D interface of 'PC Architect' when depicting the assembly of a PC to the 3D assembly interface of 'PC building simulator'. The 3D interface of 'PC building simulator' was seen as too hard to control by the users who reviewed the app.

Survey

In order to get a better understanding of the target market, I have made a questionnaire about PC building to distribute to the public and more experienced PC builders.

PC Builder Questions

Here are the questions I chose:

1. Do you have or use a PC?
   This question was to see how much of the population is familiar with a pc and use them.
   (Possible answers: Yes or No)

2. Would you build or upgrade your pc?
   I chose this question to see how many people are considering building a PC and possibly using a product like the one, I am going to develop to help them.
   (Possible answers: Yes or No)

3. Which element of building a pc do you like?
   This question is to see what features that user would enjoy and that I could integrate into the product.
   (Possible answers: Shopping for parts, Assembling the PC, Benchmarking aka testing the build, other *please specify*)

4. Which element of building a PC do you hate?
   This question is to see which features I should not include or that my product could fix and make more enjoyable.
   (Possible answers: Shopping for parts, Assembling he PC, Benchmarking aka testing the build, other *please specify*)

5. Which of these brands have you heard of?
   This Question is to give me an indication of what parts would be recognised or expected to be in product.
   (Possible answers: NVidia, AMD, Intel, Alienware, *Name any more PC building related brands*)

6. Where would you gather parts for your PC?
   This question is for the more experience PC builders and it is to give me an indication of where the more reliable and cost-efficient sources of PC parts.
   (Possible answers: online shops, local tech shops, second had from proper you know, other *Please specify*)

7. List the stats of your dream PC:
   This is to give an indication of what the users would be looking for in a PC build. It also allows me to see the price and performance levels that the users would like from the available parts.
   (Possible answers: *open answer*)

Interface Questions

1. Do you like 2D retro (pixel) games?
   This is to see if the possible users would agree with the 2D art style I found to be efficient in my existing solutions research.
   (Possible answers: Yes or No)

2. How would you like to navigate from page to page?
   This is to see how the user would like to transverse the user interface.
   (Possible answers: Tab button, banner menu, swipe gestures, other *Please specify*)

3. How much space would you like an app like this to be?
   A form that I can see my product taking is a mobile application. This question would help me know how much space the users are willing to give on their devices for a product like the one I want to develop.

(Possible answers: <5mb, 5mb to less than 15mb, 15mb to 25mb, >25mb)

4. Would you use this app?
   This question is to see what the demand is for the product I am producing.
   (Possible answers: Yes or No)

Survey results and conclusions

After filing though my returned questionnaires, I collected the results and here is what I found:

1. Do you have or use a PC?
   95% of the people said Yes in my survey as opposed to the 88% in the wider UK population n[reference]

2. Would you build or upgrade your pc?
   22% of my sample said they would. This shows that although there is a high amount of PC usage as seen by question 1, there is not a lot of demand for upgrading or building a PC.

3. What of building a pc do you like?
   Of the 22% that said they would build or upgrade their PCs;
   15% said they like shopping for parts,
   80% said they like assembling the PC,
   No one said benchmarking aka testing the build,
   The rest had a response in line with selling old parts to earn more money.
   This result shows me that the users may also like a feature that references or recreates the assembly process, which they like the most.

4. Which element of PC building do you hate?
   95% of the 22% that said they would build a PC said that testing the build (troubleshooting) is the part they hate the most about pc building. The other 5% said shopping for parts. This tells me that the main issue people have with building a PC is ensuring that it works reliably once assembled. 'pcpartpicker.com' overcame this with detailed guides of the PC builds. My product mainly helps the 5% who did not like shopping for parts because it allows them to interchange parts and see which combinations are best for them. Another useful feature could be a recommended parts section, which can offer similar parts to the one they are testing for.

5. Which of these brands have you heard of?
   All 100% said they have heard of Intel, this means that it would be beneficial to include their products.
   45% have heard of AMD and NVidia.
   30% said they had heard of Alienware
   No one recommended any more brands

6. Where would you gather parts for your PC?
   94% said they would get their parts from online shops. This leads me to believe that they would find it beneficial to have online shops integrated with the product make purchasing the parts easier.
   The remaining 6% said that they would use a local shop. This could be because second hand parts are cheaper in these shops.

7. List the stats of your dream PC:
   From those who answered the question I found that the graphics cards were a point of interests in which

most of their budget went towards. Because of this, I believe a good variety of graphics cards would help them make a better-suited decision.

Interface question results:

1. Do you like 2D retro (pixel) games?
   Not everyone understood this question but those who did were split 64% Yes to 36% No. This shows me that the majority of those who responded would be fine with the 2D interface I was planning.

2. How would you like to navigate from page to page?
   40% said banner menu, 28% said tab buttons, and 32% said swipe gestures. This may seem like the majority of people would prefer a banner menu or swipe gestures to tab buttons. However, the developers at Webx Solutions told me that it is more efficient in terms of user interaction and programming to use tab buttons. I will take their advice and use tab buttons.

3. How much space would you like an app like this to be?
   36% wanted the app to be less than 5mb with 52% accepting an app size between 5 and 15.the remaining 12% were fine with an app being more than 15mb. Because of this, I will try to set myself a limit for the prototypes to be less than 15mb in file size.

4. Would you use this app?
   98% said yes but I believe this was a bias result.

Specifications

Now that I have completed my research for the problem, I have created a set of essential features that my product must have to resolve the problem.

- The backbone of my application is a database that holds information about the parts the user is able to interact with. The use of a database allows me to retrieve the data from a persistent (organised) data store which I can integrate into the source code of the application. A database also allows me to sort, order and filter the records to fit user queries for PC parts in a sustainable way. This can expand as more PC parts added to the system.

- As I touched on the point above, there needs to be a database querying system to allow the user to access the parts that they are interested in quickly. This is better for the user experience because it means that they can quickly interchange parts they want to test.

- In terms of what the database is populated with, I will need tables contain the various PC parts that are needed to build a pc and for the user to experiment with. In order to build a PC you would expect to have a

    - Motherboard
    - CPU
    - CPU cooling
    - RAM
    - PCI GPU (Graphics card)
    - Power unit
    - Secondary storage drive (HDD, SSD etc.)
    - Case

  There needs to be a table for each of these in the database so I can make specialised tables with fields that suit the component type. This helps with normalisation and the efficiency of the database.

- The application must have a Graphical interface where the user can select, assemble and interchange parts in PC build. This the where the user can access the database and experiment with parts. It needs to be graphical interface it is an intuitive and accepted way of using applications on modern mobile devices. In addition to that, my research shed me that the 2D assembly screen from 'PC Architect' is a very efficient way to experiment with PC parts. In addition, the list view from both 'PC Architect' and 'pcpartpicker.com' showed me how effective the list view is when using it as a user interface for a database. A graphical interface can help create a better user experience because it displays a lot of information to the user about the parts in a small space that can also be sort and filters to help the user find the parts best suited for them.

- In order for the user to work on a build for a prolonged amount of time inside and outside of the application, there needs to be feature that stores the parts included in the current build so that the user can recover them and continue experimenting at another time.

- I believe that there should also be a comparison screen; this is to compare parts so that the user wants to decide between them. This will be beneficial for a user to make a quicker and better-informed decision because they no longer need to change between parts and memorise the statistics to compare parts, which is an inconvenient system.

- Finally, a major priority is that the application works smoothly, bug-free and is responsive to user request. As my research shows a bad finish on the integrity of the code ruins, the user experience and they will be unable to be productive with the application.

Limitations

I cannot accomplish some objectives with my solution this includes:

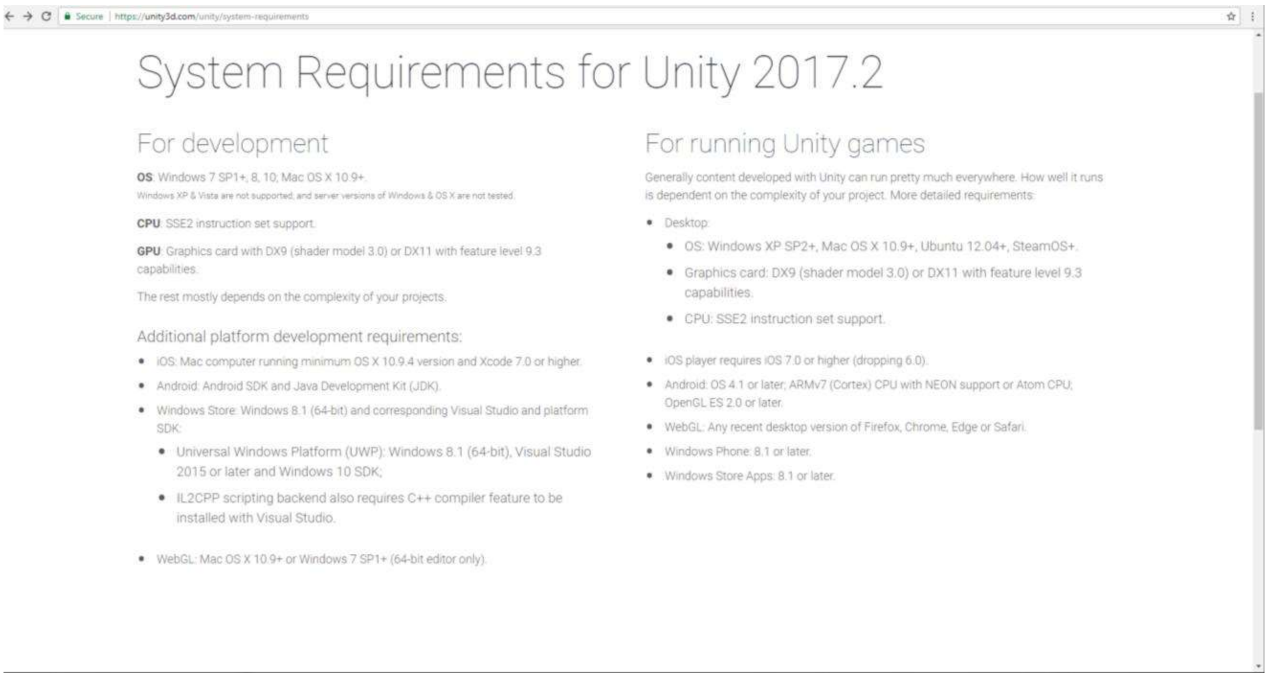
- Due to the time and resources allocated for this project, I will not be able to develop this application to have a wide variety of PC parts. Populating the database will use up time that I can spend coding, because the focus of this application is its program mechanics, I need to focus on the code first. I can expand the PC parts registered in the database once I can prove that the mechanics can handle an infinite about of PC parts in the databases. However, for a small proof of concept application, like the one I will make, variety is not a top priority, despite my research showing how important it is for the user experience.
- Another limitation is that I do not have the licensing to use real parts in my application. If I were to publish and distribute the application with real world PCs parts in the database, I will be susceptible to legal troubles from the company owners for using their products without their permission to enhance my own. I could go to all the companies like Intel and NVidia, which people know to be popular in my research because I do not have the time and budget to pay for a legal team and to sort through the paper work. Instead, my application I can take influence from 'PC Architect' and use fake or placeholder names to use as alternative PC parts to show that if they were real licenced parts my application will still work.
- Due to the Data protection act (1998), if I were to hold information about the user such as card details to perchance PC parts, I will need to ensure it is well encrypted an secure. This can be done with hashing algorithms and encrypted servers none of, which I have not access to that can satisfy the standards of the Data protection act (1998). Better security is an improvement I can make if I get the appropriate funding for the project or the resources are provided.

Project requirements

Development hardware requirements

I plan to develop this application using the Unity game engine with C# scripting. I chose the Unity engine because of its 2D support for developers. In addition to that, I see this project will help me to gain confidence in using the Unity engine in other ventures involving C based languages. Finally, the main reason I am using the Unity game engine to produce an app is that it has testing features like the Unity remote that can let me test my project on the product's native hardware as I develop. The unity engine also allows me to export my projects as executable files (.exe) for windows or as Android, IOS, Linux or any other platform that can run applications.

For my Unity projects, I use my laptop, which is HP 15-ba079sa Notebook running Unity 5.4.2f2, (64-bit). I believe this hardware is enough to build and develop the application. This is because of the comparison of my laptop's specifications compared to the Unity's recommended specifications show that it is powerful enough to smoothly run the engine. Here is a screenshot of the Unity download requirements



Secure | https://unity3d.com/unity/system-requirements

System Requirements for Unity 2017.2

For development

OS: Windows 7 SP1+, 8, 10; Mac OS X 10.9+.
Windows XP & Vista are not supported; and server versions of Windows & OS X are not tested.

CPU: SSE2 instruction set support.

GPU: Graphics card with DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.

The rest mostly depends on the complexity of your projects.

Additional platform development requirements:

- iOS: Mac computer running minimum OS X 10.9.4 version and Xcode 7.0 or higher.
- Android: Android SDK and Java Development Kit (JDK).
- Windows Store: Windows 8.1 (64-bit) and corresponding Visual Studio and platform SDK:
  - Universal Windows Platform (UWP): Windows 8.1 (64-bit), Visual Studio 2015 or later and Windows 10 SDK;
  - IL2CPP scripting backend also requires C++ compiler feature to be installed with Visual Studio.
- WebGL: Mac OS X 10.9+ or Windows 7 SP1+ (64-bit editor only).

For running Unity games

Generally content developed with Unity can run pretty much everywhere. How well it runs is dependent on the complexity of your project. More detailed requirements:

- Desktop:
  - OS: Windows XP SP2+, Mac OS X 10.9+, Ubuntu 12.04+, SteamOS+.
  - Graphics card: DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.
  - CPU: SSE2 instruction set support.
- iOS player requires iOS 7.0 or higher (dropping 6.0).
- Android: OS 4.1 or later; ARMv7 (Cortex) CPU with NEON support or Atom CPU; OpenGL ES 2.0 or later.
- WebGL: Any recent desktop version of Firefox, Chrome, Edge or Safari.
- Windows Phone: 8.1 or later.
- Windows Store Apps: 8.1 or later.

For development, the first thing that is mentioned is the operating system. Unity allow for a windows 7 and newer variation but I am using window 10 (64 bit) on laptop. Windows 10 is the most recent windows operating system and from using the engine on window 10, I have seen that it works well.

The CPU requirements are for the development hardware to have SSE2 instruction set support. This is found in most modern processors. In terms of CPU performance, I would recommend at least a quad core at 1.5 GHz clock speed or a dual core with a minimum clock speed at 2 Ghz. This is because the CPU needs to be able to run the engine at a stable rate but in the development of this application it should not be too stressful for CPUs at these levels to handle. My laptop is using an AMD A6-7310 APU; this is a quad core 2 GHz clock speed processor and can handle Unity on 2D applications.

The GPU requirements for Unity is for it to have Direct x 9 (Dx9) capabilities. The application I want to develop is not graphically intensive. My laptops graphics card is an AMD Radeon R4 Graphics card. This is a 400 MHz card with 512mb of direct memory. This is a relatively weak graphics card and I see it as a suitable minimum GPU requirement to go on to develop this project because I do not expect the project to be graphically intensive.

Unity does not specify the minimum amount of RAM and it would fit into the category of beings dependant on the project. My laptop has 4Gb of 1856Mhz RAM although only 3,45Gb is available for general use. I would recommend at least 2Gb of RAM in order to develop the application. As more features are added, RAM may become a limit in performance but it should still be useable for developing this project.

In order to download ad run unity I would recommend at least 20GB of storage space of an SSD or HDD that has at least 64GB dedicated to Unity. This is so that there is enough space for the engine itself and any project assets that to be created or downloaded, this also leaves plenty of space of the Operating system and other applications outside of the engine.

Development Software

Alongside the unity engine, I will use notepad++, which is an open sources IDE to create and edit the C# scripts that run the applications behind the Unity engine. Unity has an inbuilt console that can use to debug errors in the code. I chose notepad++ firstly because it is free to use and it can change the scripts quickly, which allow me to make corrections more easily.

Secondly, I will be using the SQLite3 plugin for Unity, which allows me to use SQL to manage and to display information from the database containing the PC parts. I chose the plugin because it is integrated internally with the C# script and it can be utilised when the project exported as a completed application.

Finally, I will use the DB browser for SQLite to monitor and validate that the database is functioning and is populated correctly.

Testing hardware and software requirements

Unity has its own built in emulator that can run the project while in the development window. In order to make sure the application also work on mobile devices I will use the Unity remote app on my own phone to test how well it works on a range of devices. My phone is a Sony Xperia X running Android 7.1.1; it fits into a modern midrange phone category. My stakeholders at Webx Solutions have also offered a range of mobile devices to test on to see how well it performs on these devices to ensure quality and a smooth user experience. From Unity's requirements they say to run games on mobile devices, they should have an operating system at a level of Android 4.1 or IOS 7. This is to ensure that the correct libraries and functions be accessed by the application from the operating system.

Success criteria

In order to make sure that my application is satisfying its aims and objectives as I develop it. I have created success criteria to reference to and to measure how well the prototypes are resolving elements of the problem. The success criteria outlines the attributes of the application that it needs to have to a suitable solution to the problem.

User experience (usability) objectives

There are two aspects of the prototypes that can be used to form measurable success criteria. The first is the user experience, which can be measured by asking the stakeholders or testers to fill out a user questionnaire are to rate how understandable and easy to use the application is. Here is what I image the user experience, success criteria to be like.

Name:			Testing group: (i.e. alpha tester, beta tester, stakeholder)		
Questions	1(Poor/terrible experience)	2	3	4	5 (very good /flawless)
How easy was it to understand the interface?					
How responsive was the application to your inputs?					
How easy is to find and select parts?					

How easy is it to move from page to page?					
How would you rate your ability to mix and experiment with PC parts?					
Rate the variety in PC parts to select from					
How would you rate the information you were given for each part when selecting?					
Total score: Notes:					

The questionnaire is structured with a name and testing group section at the top and the success criteria point is listed on the left column. Each point has a scale of 1 to 5 to rank how well the prototype meets the success criteria from the user's point of view. A rating of 1 being the worst or a terrible experience with that point and 5 being the best or flawless experience. Decomposing the user experience into these pints allows me to see which elements need to be worked on independently to improve the user experiences as a whole. I left a notes section at the bottom for any recommendations or information about their test. I will deem the success criteria met if the average rating of each point is between 4 and 5 after a prototype is released, tested and reviewed.

Application content Objectives

The next part of the success criteria is the objectives that the application musts be capable of in order to solve the solution. The success of the application to this set of criteria can be measured in the form of a checklist. As the points are satisfied to a suitable standard judged by the testers, it can be checked off the list.

Success criteria points:

- The application must visually represent a PC building experience
- The application must be able to retrieve data from the databases
- The application must display the retrieved database information in a user friendly way
- The application must allow the user to add PC parts to their build
- The application must allow the user to remove parts from their build
- The application must be able to limit the number of parts involved in the build to a logical amount
- The application must be able to save PC builds
- The application must be able to load saved PC builds
- The application must be able to delete saved PC builds
- The application must allow the user to query the database in convenient ways
- The application must be able to display information about the PC build
- The application must be able to recognise parts that are compatible with each other from parts that are not
- The application must transition between pages to enable different functions within the application
- The application must be able to close (exit) on demand

I believe these points are achievable with the resources and time I have been given. They provide enough functionality to able the user to experiment with PC parts and make a better-informed decision of what parts they want in the real world.

Agile project development cycle

The basis of my development of this application is an agile project management cycle. Agile project management is a development approach that is centred on continuous improvement, schedule flexibility and prototyping through an iterative development cycle that allows the product aims to change.

Like with all project development cycles there is a vision stage that that the process, in which the idea of project is first thought. That is this analysis section of this document. From here a design section will begin, in which the project is broken into modules though decomposition as discussed earlier. These modules go through a design process to plan the approach, resources and program logic that will be coded. The plan is then put into effect and debugged before being given to users and quality testing. Feedback from this is take into another iteration in which the required changes from the feedback is matched with the success criteria to evaluate the effectiveness of the prototype and to see which changes need to be made. This continues until both the testers and I am satisfied the success criteria above has been met and the module is complete. The next module is then developed and combined with the other completed modules. The system repeats until all modules from the original design and those created from feedback are completed. At fit point, final testing will be done to look at all aspects of the product and a final review is made. If the product is deemed to be of the quality to be released to the client or published it is done.

Analysis bibliography

[1] SOS Survival Escape Simulator. (March 18, 2017). PC BUILDING SIMULATOR.
Available: https://play.google.com/store/apps/details?id=com.pewpaf.computersimulator&hl=en (Google Play store)
Last accessed 30 sept 2017.

[2] Games From Garage. (May 2016). PC ARCHITECT.
Available: https://play.google.com/store/apps/details?id=com.GamesFromGarage.PCArchitect&hl=en (Google Play store)
Last accessed 30 sep 2017.

[3] no known creator. (N/A). PC PART PICKER WEBSITE.
Available: https://uk.pcpartpicker.com/
Last accessed 30 sep 2017.

[4] Unity Technologies. (Original release June 8, 2005). UNITY GAME ENGINE.
Available: https://unity3d.com/
Last accessed 30 sep 2017.

[5] Unity Technologies. (Last updated August 29, 2016). SYSTEM REQUIREMENTS.
Available: https://unity3d.com/unity/system-requirements
Last accessed 30 sep 2017.

[6] SQLite Consortium. (Last updated August 24 2017). SQLITE.
Available: https://www.sqlite.org/
Last accessed 30 sep 2017.

[7] Mauricio Piacentini. (Last updated September 20 2017). SQLITE BROWSER.
Available: http://sqlitebrowser.org/
Last accessed 30 sep 2017.

[8] Don Ho. (Last updated August 30 2017). NOTEPAD++.
Available: https://notepad-plus-plus.org/
Last accessed 30 sep 2017.

[9] Sony. (Original release May 2016). SONY XPERIA X PHONE SPECIFICATIONS.
Available: https://www.sonymobile.com/global-en/products/phones/xperia-x/specifications/
Last accessed 30 sep 2017.

Design

In order to design and go on to create a solution, I have gone through a process of decomposition of the problem and ways to solve it. My decomposition approaches the solution from the perspective of the content of 3 pages in the application that resolve different aspects of the problem to provide a solution. The initial layers will be for the user interface and the deeper layers the tree the for the problems in the 'backend' of the program (i.e. file & database management and source code).

Decomposition tree

Tree with annotated explanations