

Computing Coursework 2018

Planning

Initial transcript

Client

so basically, Crypto Exchanges have APIs

I was wondering if it would be possible to create a desktop app that collates all of these into one manageable portfolio

I cannot find a windows PC version of any manager out there

and certainly not one that imports using the Apis provided by the exchanges

Me

hmm like information on the current exchange rate?

Client

yeah, and pulls the current amount of stock you hold in each coin

bittrex currently have one that I can use on an iOS app

Me

hmmm interesting - I mean it would need to integrate with wallets which would be more complex -> though why not just use a website to look up this stuff?

Client

I have 5 different exchanges

about 10 coins on each,

keeping the value of each and the percentage profit is a nightmare

especially if I'm day trading

I just need a better way of keeping track

Me

hmmm okay

would be interesting to work on - let me just have a look at the apis out there

Client:

Alrighty

Me:

so I just got bittrex on my phone and I see the market you mean - you sure there's no one of these for windows already?

Client:

They provide an API, which I have found only one app that can use it

There's one company called Delta which could potentially be releasing something
Just wondering what your thoughts on the whole situation were

Me:

I saw hmmm - i'll have a look at making a simple PoC and see how long it takes to integrate stuff together - looks like a fun project - and delta looks pretty good - I'm surprised no ones released a desktop version...

Client:

So am I, I would have thought they would release desktop before iOS or apps

Me:

mhm

I guess mobile is such a big market atm?

[...]

Design transcript

[...]

Client Brief

After the initial trial commit the client requires a more visible version of the product

A desktop application will allow me to view my current portfolios and balances of bitcoins and various other cryptocurrencies. I would like it to automatically update with the current market price of the bitcoin to other currencies. I would like it to be customisable, stylish and easy to use. Additionally, I want it integrated with as many different currency exchanges as possible to maximise its usage.

MVP Plan

Minimum viable product - my initial requirements of the client's requirements

```
1  !e "tvtr a ee attlncaivi
#  Arnlnty tv ma"e a tvr$vlrv
   a  Arnlnty tv aee a %allet&e'chai(e& nmtle amvuitv cvni
      n  Arnlnty tv remvie %allet &chai(e nmtle amvuitv nlnialcvni
   r  Arnlnty tv %atch cvni (ani & all relaiie tv the nlnialnitut
)  *vv"ut curreite'chai(e rate
   a  +ut tvrt vr mulitle e'chai(e
      n  Aiera(e
      nn  ,niaice
      nnn  ,n$lyer
      ni  ,n$nie'
      i  ,nthumr
      in  ,nt amt
      inn  ,n-re'
      innn  .vnitie t
      n'  .vnivie
      '  /ea'
      'n  /emini
```

```

'nn Ontrtc
'nnn 1vrrnt
'ni 1ra"ei
'i *nqun
'in 2vlvine'
'inn 345
r Allv%ni (chai(ni (lvcacurrencyvier nvi
6 +ecumty
a /vv(le Accvuit ra ee
n 7%v actvr auth
nn 2a %vre
8 Aialyic
a 7hn n tv aialy e %hat acivi haie reei ta"ei ni the attlncaivi9
: *ncei ni(
a 7he cleitha u((e tee hevily %n he the attlncaivi tv e'n t9 Oe %vule re %nlni( tv
teiemviey vr nt9 7hvu(h ha aeenivially niencatee that nt %vule re ; ie tv ell vi9 <vr
thn requiremeit = %vule ieee tv nitrveuce a lncei e erier v = cai cvitrvi %hv n
authvrn ee & ha tane vr the attlncaivi aie %hv ha i't9

```

Similar product research

=i the aim v ma"ni(my attlncaivi the mv t releiait aie tv ivt reniieit niieitee trveuct 9 =
lvv"ee at maiy mmilar trveuct acrv en>ereit tla\$vrn p

Coin Ticker iPhone -

<https://itunes.apple.com/gb/app/coin-ticker-bitcoin-altcoin/id636476147?mt=8>

.vni ic"er vr n2hvie trvinee maiy eature

Cryptolio - <https://github.com/larion/cryptolio>

Terminal ra ee cryttv currency tvr\$vlrv

CryptoCompare - <https://www.cryptocompare.com/portfolio/>

General development model

7hrvu(hvut the eeielvtmeit v thn attlncaivi =haie vttee tv chvv e a tnral mveel v
eeielvtmeit9 7hn allv% me tv create a iery eetanlee tlai tv hv% the %vr" ieece ary tv the
cvur e%vr" requiremeit aie aeenivially reni(arle tv eeielvt the re tattlncaivi tv nrle eumi(
the hvrt eeielvtmeit %niev%9=t al v allv% me tv eialuate my attlncaivi ter vrmaice at the
eie v the eeielvtmeit chai(e9

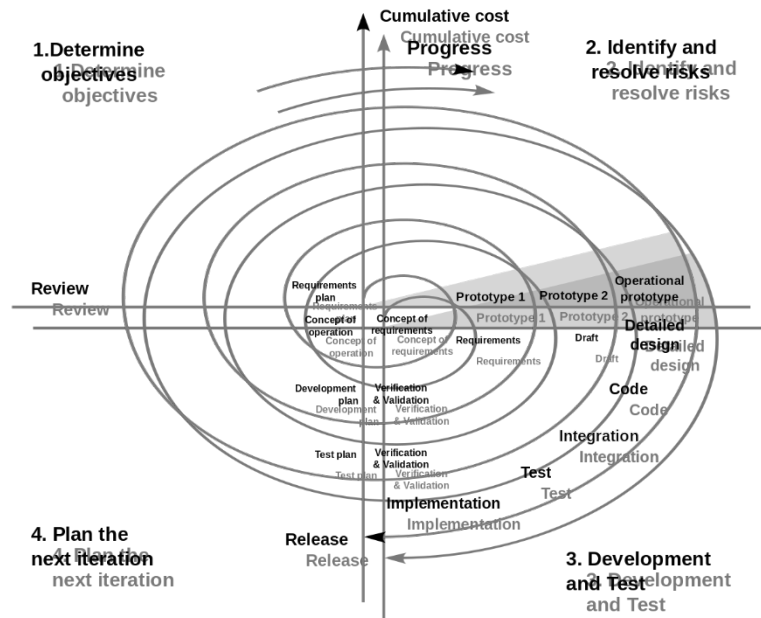


Figure 1. Spiral model development

Technologies needed

Language Choice

There are many languages available that can be used to develop the system. The choice of language is a critical decision that can affect the success of the project. The language should be chosen based on the requirements of the project, the skill set of the development team, and the availability of resources. The language should also be chosen based on the performance requirements of the system, the security requirements of the system, and the maintainability requirements of the system.

C# / WPF - <https://cocosm.com/development/development-technologies/development-technologies-1/>

The language should be chosen based on the requirements of the project, the skill set of the development team, and the availability of resources. The language should also be chosen based on the performance requirements of the system, the security requirements of the system, and the maintainability requirements of the system.

Advantages

- + The language is easy to learn and use.
- + The language is powerful and flexible.

Disadvantages

- The language is not as fast as other languages.
- The language is not as secure as other languages.
- The language is not as easy to maintain as other languages.

Figure 2. Spiral model development

The language should be chosen based on the requirements of the project, the skill set of the development team, and the availability of resources. The language should also be chosen based on the performance requirements of the system, the security requirements of the system, and the maintainability requirements of the system.

Advantages

- + The language is easy to learn and use.

Disadvantages

- The language is not as fast as other languages.
- The language is not as secure as other languages.

- .lv ee +vurce
- Cvtvrvu ly rae eentvr vr the A=lv%eier nmtrviee receitly
- Haia ha lvi(cvmtle ime %hch ma"e ratne eeelvtmeit hareer eiei %nth vi the rui
cla %attni(

\$ctoon - [https://e\\$ctoon\\$oo/](https://e$ctoon$oo/)

7hn rame%vr" ceitre arvuie reni(cvmtletely crv tla\$vrmaie?u t trvineni(ni e>ect a
chrvmum rrv% er %niev% aianlarle tv reieer aiymveeri 07J *&. ++&Haia+cmitt9 [. =7A7=BC
4le1GEl #F8G]

Advantages

- + Kery ea y tv etut
- + .rv tla\$vrma
- + .ai illacce lv%erleiel B+ eature
- + <amlnar techivlv(ne
- + Btei +vurce

Disadvantages

- Oa lar(e LAJ vierheae [. =7A7=BC Kar1: El #F8G]
- *ar(er ; le nNe [. =7A7=BC Kar1: El #F8G]
- Oareer tv ma"e lvv" iaie lcaivt iaturally emree iaie A=element

Conclusion

=i the eie = relneie 4lectrvih+ n the re tchvnce tv rearle tv runle the attlncaivi the dneit iee 9
7hn n eue tvnt lv%learini(vierheae aie ea y crv Mtl\$vrmaie cvmtai rnlty9 7hn %nll re nmtvrtait
a alv%learini(vierheae ei ure the re tcvve cai re %m-ei qunc"ly aie eOcnitly9 AeeniviallyP
ni ai a(e nth a teraie a ter cvmtuter P the vlcallee Qrlvat' %e (et rvm emreeeni(e>eci iely a
chrvm rrv% er %nthni vur attlncaivi n mni(atee9 7hn n e tecially true a vur attlncaivi' mv t
nitei nie ta " %nth uievurteely etchni(eata rvm ai A2=- %hch n uiln"ely tv lv% ev%i the
%hvl cvmtuter9

APIs

Le earchni(the A2= P= %n h tv u e tv (et each rntv eata uch a curreicy rate &cryptvcurreicy
e'chai(e rate etc9 Oere' vme=haie vuie eumi(the tlaiini(ta(ep

- [h-tp&&:er%v&](https://p&&:er%v&)

Boilerplate comparison

3hei creai i(ee "tv t attlncaivi %nth electrv i there cai re a lvt v etut ni term v trvterly
n vlai i(the reieerer rvm the mani trvce Itv treieit vther trv(ram ni?eci i(cvee 9
AeeniviallyP nt n helt ultv u ea J K. rame%vr" uch a LeactH+ vr Ai(ular tv nmtrvie
eeielvtmeit ime aie treieit rul" ni the html cveera e9 7hei there' the trvlem v maia(ni(
tate ni lar(e trv(ram %hch n (eierally evie thrvu(h lnrrame ln"e reeu' %hch haie enrect
rnieni(nitv Ai(ular vr Leact l ee reeu' Mreact 9

Bie %ell "iv%i re vurce vr electrv i rvlertlate n the Ra%e vmellelectrv i S retv ntvry %hch ln t
tvvl that u e electrv i P tvvl vr electrv i P a %ell a rvlertlate p

[h- t p&&\(nthur%vcm& niere vrhu &a%e vmellelectrv i@rvlertlate](https://t p&&(nthur%vcm& niere vrhu &a%e vmellelectrv i@rvlertlate)

Electron is a framework for building cross-platform desktop applications with web technologies. It is based on Chromium and Node.js. Electron allows you to use any web technology to build desktop applications. It is a great choice for building desktop applications that require a rich user interface and fast performance.

Style choices

Testing framework

Hardware and software requirements

The hardware and software requirements for Electron are as follows. The minimum hardware requirements are 4GB of RAM and 10GB of free disk space. The minimum software requirements are Node.js 12.0.0 or later and Chromium 78.0 or later.

The recommended hardware and software requirements for Electron are as follows. The recommended hardware requirements are 8GB of RAM and 20GB of free disk space. The recommended software requirements are Node.js 14.0.0 or later and Chromium 80.0 or later.

Supported Platforms

Following platforms are supported by Electron:

macOS

Only 64bit binaries are provided for macOS, and the minimum macOS version supported is macOS 10.9.

Windows

Windows 7 and later are supported, older operating systems are not supported (and do not work).

Both `x86` and `x64` binaries are provided for Windows. Please note, the `ARM` version of Windows is not supported for now.

Linux

The prebuilt `x86` and `x64` binaries of Electron are built on Ubuntu 12.04, the `arm` binary is built against ARM v7 with hard-float ABI and NEON for Debian Wheezy.

Whether the prebuilt binary can run on a distribution depends on whether the distribution includes the libraries that Electron is linked to on the building platform, so only Ubuntu 12.04 is guaranteed to work, but following platforms are also verified to be able to run the prebuilt binaries of Electron:

- Ubuntu 12.04 and later
- Fedora 21
- Debian 8

Figure 1. Supported systems for Electron

Electron is a framework for building cross-platform desktop applications with web technologies. It is based on Chromium and Node.js. Electron allows you to use any web technology to build desktop applications. It is a great choice for building desktop applications that require a rich user interface and fast performance.

Conclusion

Electron is a great choice for building desktop applications that require a rich user interface and fast performance.

Electron is a great choice for building desktop applications that require a rich user interface and fast performance.

Basic Layout design

= ee n(iee a ra nc vierine% v %hat = %aitee the att tv lvv" ln"e %hnh n hv%i relv%9

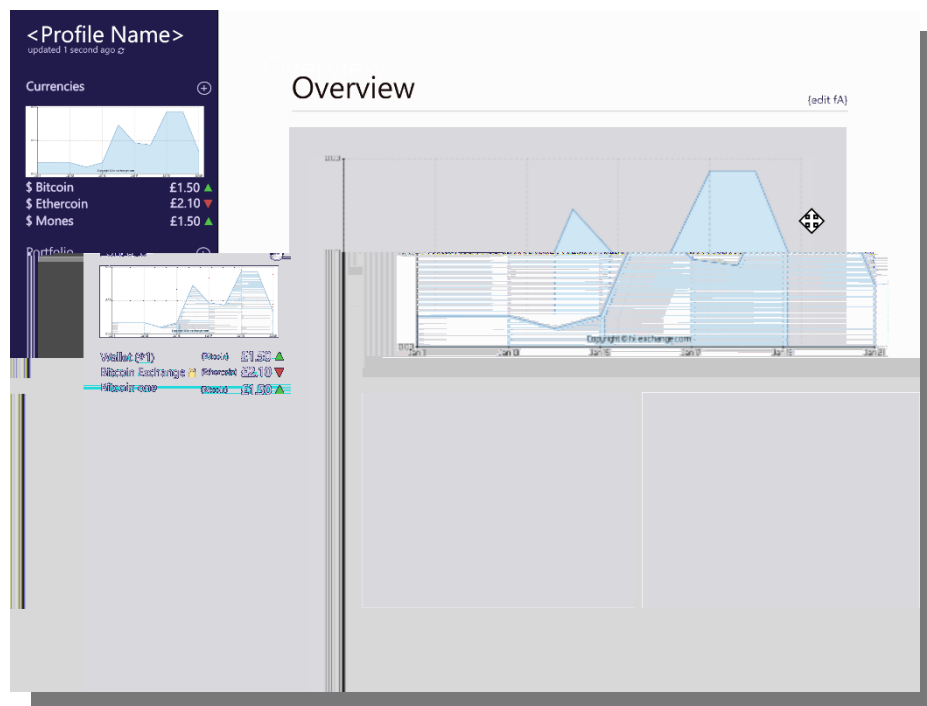


figure 2.3A basic design of what two applications might look like

. vlvr u ee vr mvc"utp

Area	. vlvr l@Oe'
*ef nee rar rac"(rvu ie	@1. 1G68
At arrv% lef neerar vrecvlvr	@6A, <6F
! v%i arrv% lef neerar vrecvlvr	@, <6#6F
2aelvc" lef nee cvlvr vrecvlvr	@<<4)G<
7e' t cvlvr lef neerar	@! G. ! <#
, ac"(rvu ie cvlvr mani area rlvc"	@! G! G! ,

7hn ee n(i n heainly ur?ect tv chai(ea the att n tu hee thruv(h eeielvtmeit9

Aeen i viallyP = mveellee ai ncvi vr the att lha i vi ra ee vi the 3n"nmeena cryttvcurre icy lv(v a hv%i relv%p

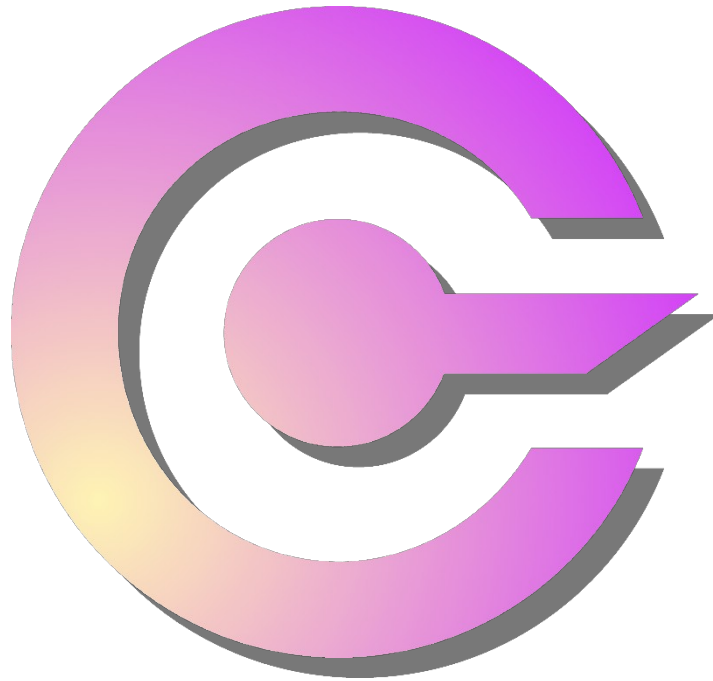


Figure 4. Etoway cryptocurrency logo 5' e Application logo

.vlvur+tecn; caivi vrlv(vp

Area	.vlvur l@Oe'
7vt m(ht nee (raeneit tv	@<<8#48
,v-vmlef nee (raeneit tv	@<: ! #6#

Tests needed for MVP

7e t iame	7e t! e cmtivi	J K2 +tec
, a nc *vae	7he attlncaivi lva e ut	
A=4'n t	7he A=n tre eit ni the reieeree attlncaivi	

Name Choice

7hn may eem ln"e a trninal ta " vrai attlncaivi9 Ov%eierP nt cvule re ar(uee that the iame ha ai
nmtact vi the clneit ine% vi the ; ial trveuct9

.vi neeree iame ieee tv reTect the iature v the attlncaivi reni(p

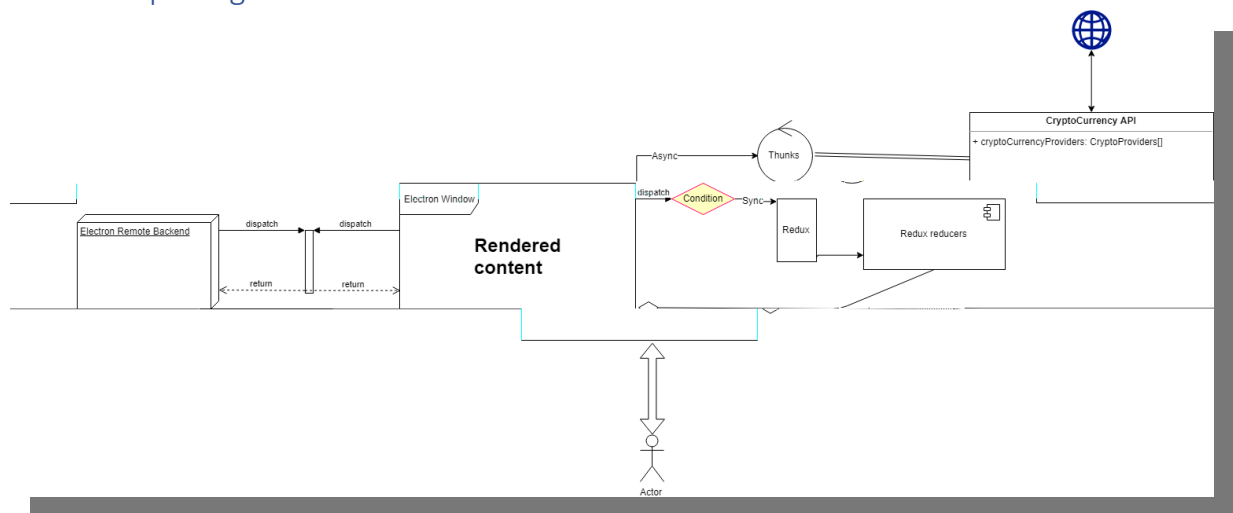
- J veeri
- .ryttvcurreicy
- 2vr\$vlrv
- +lee"
- 4a ytv u e
- +ecure
- +a e

.vi neeree iame p

- .ryttvlnv
 - 2vrtmaitua are clnchUaie iviMmveeri rute>eciie
 - CAJ 4 . *A+Op [h-t p&&\(nthur%cvm&lamvi&cryptvlnv](#)
- .ryttv,ueey
 - Bierly meielyP eve i't eem ecureV
 - CAJ 4 . *A+Op [h-t p&&%%%mycryptvruueey%cvm&](#)
- ,nt2vr\$vlmv
 - =mtlne vily vr rntcvni -vr re t erie rntcvni9

=i the eie =eeceeee .ryttvlnv vuieeee the re thv%eier nt hae a iame cla h %nth a terminal ra ee
 crypttv curreicy tvr\$vlmv9 +vP =eeceeee tv chai (e nt ln(htly n i tv .ryttvlnum9 3hnh ma"e nt vuie
 mvre trv e nvialaiea e>eciie9

Problem splitting



Development

Testing

Testing Needed

Evaluation

Testing

Bibliography

, vehmP #FF69 ile /piral model 7(ewm, !811?svg :; ikimedia .ommoos. [Bilnie]
 Aianlarle atp [h-t p&&cvmv i 9%n"nmeena9vr\(&%n"n&<nlep+tnralWmveelWI, vehmPW1XDD 9 i\(](#)
 [Acce ee F6 F# #F1D]9

4lectrv i H+P #F1G9 electroo < (uild cross pla torm desktop apps itw -ava 'cript, >%?@, aod \$""..
 [Bilnie]
 Aianlarle atp [h-t p&&electrv i? 9vr\(&](#)

4lectrvih+P #F1D9 ūported Aatorms <Aectroo. [Bilnie]
Aianlarle atp h-t p&&electrvi? 9vr(&evc &tutvrnal& uttvrteelmtla\$vrn
[Acce ee F6 F# #F1D]9

/nthur =icP #F1D9 Źeodiog \$Brepositories oo CŹt >ub today. [Bilnie]
Aianlarle atp h-t p&&(nthur9cvm&treieni (&cY#)
[Acce ee F6 F# #F1D]9

Karnvu P #F1:9 Ask > 'ŉ; wy 5; wy 'ot Źe AectrooE<>acker 'e s. [Bilnie]
Aianlarle atp h-t p&&ie% 9ycvmrniatvr9cvm&ntemVneZ1#11X#GD
[Acce ee F6 F# #F1G]9

3). P #F189 CŹaceful degrada too versus progressive eowaocemeot .; 2\$; iki. [Bilnie]
Aianlarle atp h-t p&&%%%9%)9vr(&%n"n&/race ulWee(raeaiviWier u Wtrv(re nieWeihaicemeit
[Acce ee F6 F# #F1D]9