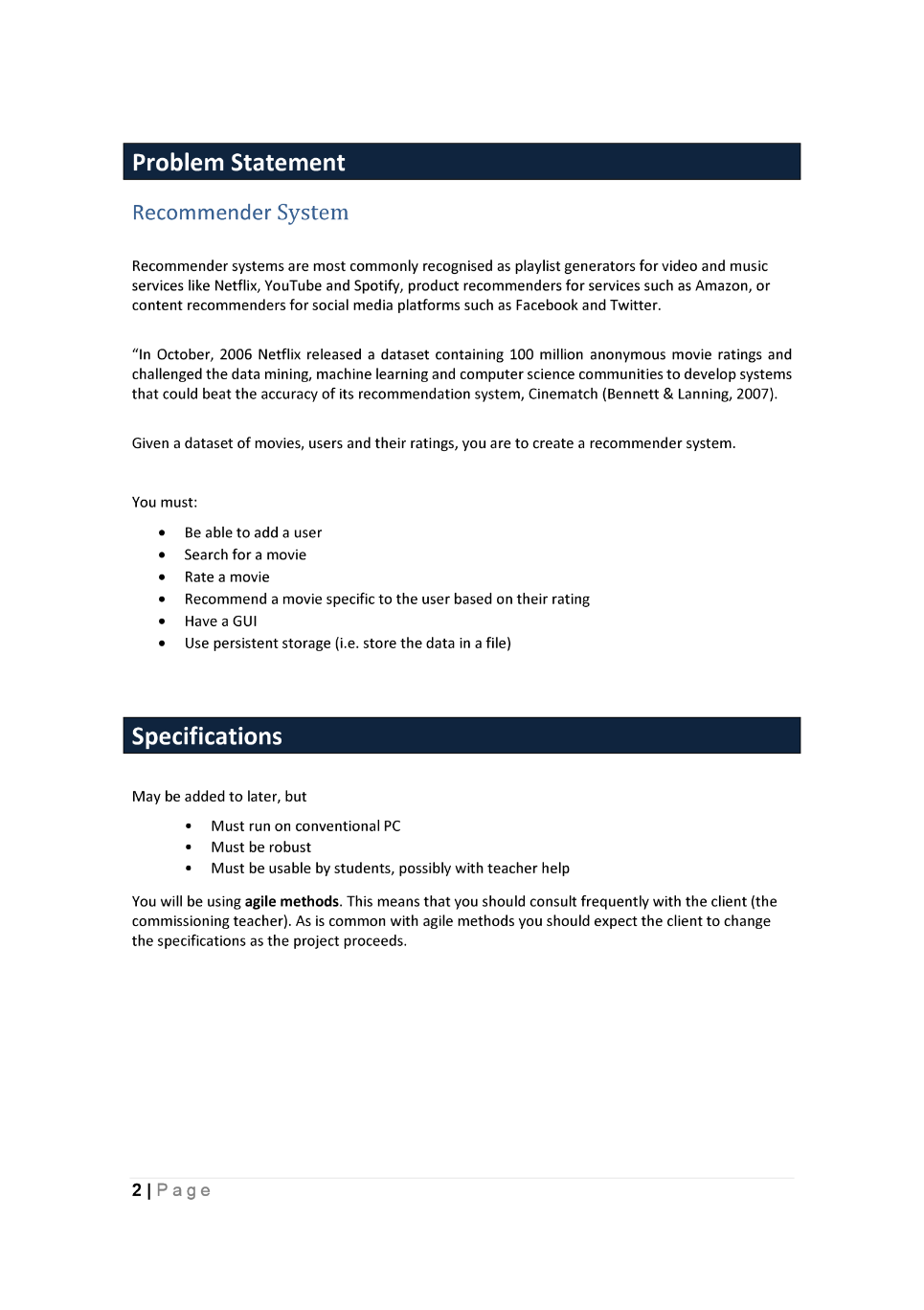
Brief



Materials and components.

Languages

Scratch

Scratch is a very visual language. It is based off of JavaScript and can be used to efficiently create quick and easy programs. Images are easy to make and use, movement is easy and manipulation of code is easy as well. I also know the language well.

But where it leads in simplicity it lacks in functionality, its text input and output is relatively simple and it’s more complex lists and functions are lacklustre in the version we have on the school computers.

Python

Python is an easy text based program. Its structure makes it very easy to read as it is a very high level language I know the language very well and so does my teacher and classmates so any feedback I get from them they will have the limitations of the language in mind. It is extremely popular online and so there is a lot of resources to help.

As it is high level it is slow in more complex functions but as it is only a text based game this is not really a problem.

JavaScript

JavaScript is an extremely versatile language as it can be integrated into HTML and CSS which is what most of the internet uses to display information. As it can be integrated into CSS and HTML it is very customisable and has a lot of room to make anything. JavaScript’s libraries make the possibilities essentially endless (p5.js). Most of the internet uses it so there are a lot of resources available to help me.

While I know js well I do not know how to integrate it into HTML as confidently as I would like. While it is definitely something I could easily learn it adds another variable and stress into the mix where I want the only thing to limit my ideas is the stakeholders.

HTML and CSS

Most of the internet uses it so there are a lot of resources available to help me. It could be a very easy way to sketch ideas and segment ideas for very simple feedback, development and isolation of bugs.

Without js it cannot be anywhere near as complex or interesting as I would like it to be. And I have not used it in a long time so I would have to relearn most of it.

Final language

As I and many other know python so well I am going to use python, its downsides are far too high level for me to comprehend let alone consider, where the upsides are perfect for the briefs specification of a text based game.

Components

Functions

|  |  |  |
| --- | --- | --- |
| name | input | output |
| Print | String, end | Prints the string to the console and then prints end |
| For | A range to go through | It goes through the range and changes a variable predictably |
| While | Bool / statement | Repeats code inside until bool/statement == false |
| Def | String, parameters and code | Makes the string into a function that runs the given code |
| If | Bool/statement | Runs the code if bool/statement evaluates to true |
| Elif | Bool/statement | Runs the code if bool/statement evaluates to true and no (el)if statement has run before it |
| else |  | Runs the code if no (el)if statement has run before it |
| In | Variables | Checks if the first variable is contained in the second and returns true if so |
| Exec | String | Runs the string as if it were code |
| Format | String | Puts the string into the string this function is applied to |
| Sleep | Integer | Stops the code for the integer amount of seconds |
| Type | Variable | Returns the type of variable it is |
| Input | String | Requests the user for input when presented with the given string returns that input |
| Lower | String | Makes all letters lowercase |
| = | Variable(s) piece(s) of data | Redefines the variables with the pieces of data |
| == | Variable(s) and/or piece(s) of data | Compares the given vales to see if they are the same and returns true if so |
| != | Variable(s) and/or piece(s) of data | Compares the given vales to see if they are the same and returns false if so |
| < | Variable(s) and/or piece(s) of data | Compares the given values to see if the first is greater than the other if so it returns false |
| > | Variable(s) and/or piece(s) of data | Compares the given values to see if the first is greater than the other if so it returns true |
| <= | Variable(s) and/or piece(s) of data | Compares the given values to see if the first is less than or = to the other if so it returns true |
| >= | Variable(s) and/or piece(s) of data | Compares the given values to see if the first is greater than or = to the other if so it returns true |

Data types

|  |  |
| --- | --- |
| Dict | It stores data with a string assigned to it |
| List | It stores data with a predictable order and numbers assigned to it |
| 2d, 3d etc list | A list with multiple lists inside the list. (and potentially multiple lists inside those lists etc.) |
| Library | It stores functions in a .py file |
| .txt | It stores a long string |
| variables | It stores a single piece of data |
| Class | It stores class wide data, instance data, methods |
| Stack | It sores data and the last entry is the first returned |
| Node | It depends on the type. But in general it is a data point with a pointer to another node, it can be more complicated. |

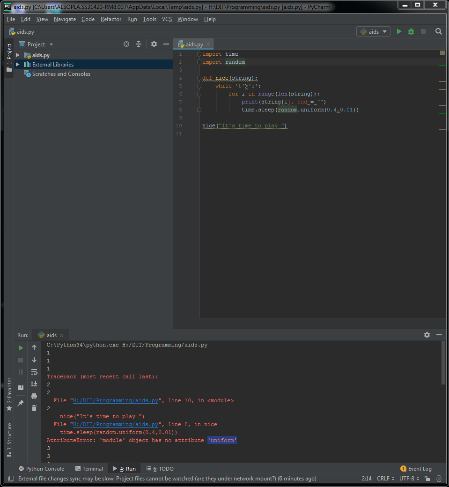
Tools

PyCharm

Looking at most internet reviews, they don’t really have anything functionally bad to say. It’s auto completion is great, it’s file browsing and navigation is superior to everything else, it GUI is great and any modules that are not on your system that you wish to import it can do it for you.

The only downside I can find is that it is slow, it depends on the computer mostly so I will attempt to install it on the school computers(or possibly onto a USB to put into the school computers) and see if it runs as badly as some people say.

After installing it on the school computers.



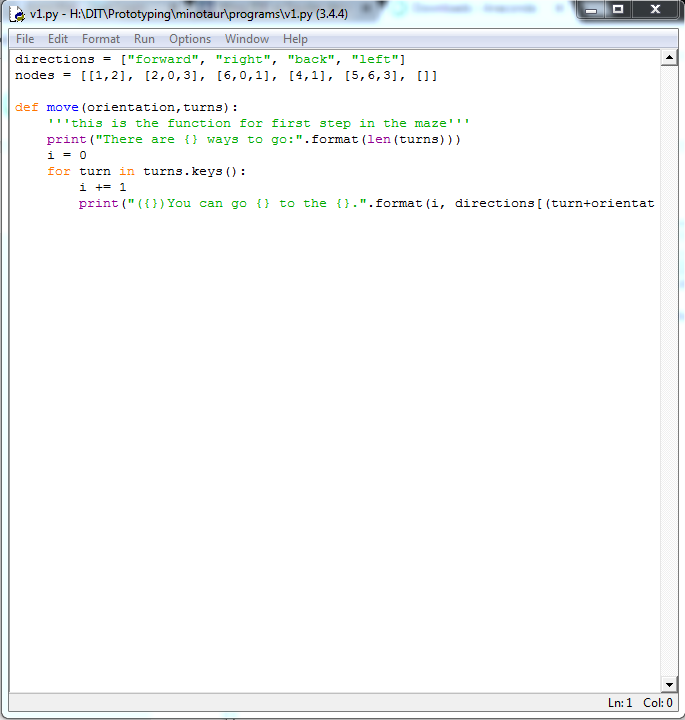
There were some issues. After trying to simply import native libraries from python I had some trouble. I couldn’t import the random library and when I tried to run the code it didn’t print anything but weird numbers that I have no idea what it means.

While I’m sure I could learn what it does and why it is doing it I just don’t have the time to learn how to use the IDE.

IDLE

After using IDLE on the school computers and at home for a while I know that it is well integrated with python with an easy console. The colours are easy to recognise and use, partly because I have already been using them and partly because they are well designed. The main reason why I would want to use IDLE is because I know how to use it and it will not inhibit me in my creative process.

File navigation is clunky at best and debugging can be a pain with no line numbering.



Wing IDE (101/personal)

Wing IDE is said to have a good debugger, easy shortcut programmability, macro capabilities and many plug-ins.

But it is at a price, $39 per user and so even if it was a brilliant piece of software, a price wall immediately removes the possibility of me using it as I haven’t even tried it and I don’t have time to evaluate this purchase let alone learn how to use it.

So I will look at its little brothers, wing 101 and wing personal. After looking on the internet I could not find any in depth reviews on wing 101 and personal and as I could not install it as a user on the school computers I cannot review them for myself.

Spyder

Spyder uses a powerful library for auto completion called rope. It is free which a must for this project as. It can plot graphs but this will not really be useful for the user but could be useful for path analysis in the maze and easily viewable data on pathways and potential changes to the pathways depending on the stakeholder’s feedback. It works on Linux, Mac and Windows. It is lightweight which is good for our relatively slow school computers. And it is solely designed for python which means it is optimised solely for python.

It is ugly, and it is not as good at debugging as other IDEs.

After installing it and using it I really like it, there was no issue with importing libraries, the biggest downside, the debugging is great (better than IDLE the only other IDE that was working for me) as it has line numbers on the side and you can run it in the program itself.

One downside was that the time.sleep function is not as good as IDLE but that could be because I am running it natively in the program and not in the separate console, and even if that is an issue I can run it in IDLE and code it in Spyder. The one last test is if I restart it and it stays on the school computer as every time it restarts it resets the data and if the program is not still on the computer I will not be willing to reinstall it 3 times a week.

Final IDE

Sadly I have to choose IDLE for my prototyping IDE, this is because while it is not the best IDE for the job it is the only one that can be used efficiently on the school computers and any IDE that is online is no good either as my internet is shut down at 9:00pm (that is why I have not looked at any online IDEs).

VCS

GitHub

It is in a simple GUI and so you don’t need to learn anything complicated to use the repository, as it is so widespread it has a lot of documentation for beginners and allows for everything GitHub has to offer to be accessible. It is a simple backup that allows me to access it wherever I am, school, home, otherwise. Security, however a high schoolers project doesn’t need to be that secure, some features are locked behind a paywall, which given I am not planning to invest in GitHub permanently (currently) however private repositories are still free, so it doesn’t really matter. It has Kanban built in.

Gitlab

Easy to work with, restrictions are easy to implement, it has its own task manager but it is messy and sometimes a pin to work with. Backing up is annoying at times. The help system is automated and useful. Navigating the web UI is annoying, as this is something I will be doing, this stops me from using it.

Bit bucket

It is simple, and in turn easy to use, the UI however is clunky and old, larger projects tend to slow the system, but as I will be creating a simple AI this will not be much of an issue. Many features are behind a paywall, immediately stopping me form using it.

Overall I will be choosing Github purely because it is the industry standard, and so I will most likely be using it next year, the UI is better than both bitbucket and gitlab and this does make me want to use it more. It is industry standard and I will use it next year, the better UI, and industry benchmark in every aspect is just a bonus.

Equipment

I am looking to use a keyboard, mouse and monitor to interact with the program, while it will be mostly keyboard and monitor the mouse will be used to navigate to the program and start it.

While I could use a tablet and phone and I could potentially look into mobile options that will be dependent on if I can find a python program that can import files and of course, stakeholder feedback. When it comes to the prototype there is no question that keyboard and mouse will be far more efficient than the touchscreen keyboards available.

Not to mention the fact that I am most comfortable coding python and JavaScript and do not have much experience coding in languages that support other platforms, so I am almost limited to desktop.

Stakeholders

* Mr Ny
* Other students
* Family

Students, teachers and parents are all involved in exams, and so getting a sufficient range of feedback from all of these is required to make the most effective program possible.

Health and Safety.

To remain healthy and safe during the creation of this prototype there are certain actions that must be taken.

* I will sit up straight to prevent back injuries form prolonged desk work
  + This involves my head being directly above my neck base to minimise the stress on the neck
  + Making full use of my seat and its lumbar support
  + Moving my chair in as far as possible to support my arms with the desk
  + Ensuring the monitor is directly in front of me to not twist my neck unessecerily
* I will have the seat height so that my elbows are at 90° and ensuring that my hands are equally distant from their corresponding shoulder
* I will avoid looking at a monitor for more than an hour and sitting at a desk for more than an hour and if I am planning to do more than an hour session, plan for regular breaks with water and food when needed
  + Rested eyes function better [[1]](#footnote-1)
  + A hydrated brain functions better[[2]](#footnote-2)

Prototyping Techniques/Processes

Story-boarding

Story-boarding is a visual representation of how events will occur in your prototype.

It consists of simple pictures that show what will happen, when and where it will happen and how it will happen.

Given my program isn’t really going to have much flow to it, it will merely be from page to page or buttons to button there isn’t much use for storyboarding and wizard of OZ is more suited to my current task.

Rapid Prototyping

Rapid prototyping consists of three simple stages, prototyping, review, refinement and repeat.

At first a prototype is made this can be a simple prototype or a complex one, it can be a piece of paper or a 50GB program.

Then the prototype is reviewed by the stakeholders, they share what they like, what they don’t and any new ideas they have for the product/prototype.

Then using the feedback from the review the prototype is refined (or possibly scraped and rebuilt) and then the cycle repeats.

As this is the foundation of the entire idea of prototyping it is something I will use in practically every single iteration and for practically every single decision I make after the initial idea.

Wizard of Oz Prototyping

Wizard of Oz prototyping is when you use an analogue program to represent a digital one. For example if it was a website then there would be multiple real pages of paper to represent the possible webpages, usually this is done for visual programs where it is very expensive to make complex visuals compared to the cost of the pens and paper to create them.

As my prototype is going to be a series of pages, or frames on a page wizard of OZ will be useful to determine what the users find the most easy and useful, and just the best when using my program. And given the program is so simple I can just draw out changes that the stakeholders want or let them create the pages and let me program what they want to be as the interface.

Agile

The methodologies behind agile is to solve the typical problem of software and stakeholders tending to change their requirements over time as they see the product being developed or hear new perspectives on the software they develop their requirements will change. This is much like prototyping, but instead of just doing it for a fast simple version, it is a permanent design philosophy. Agile is designed to have a constant loop where requirement changes are an integral part of the development process this does result in a slowing of more traditional projects, ones that only have one unchanging goal, but as few real modern projects are like this agile is used in most projects today. As the agile concept is just that, a concept. There are many manifestations of this philosophy that aim to achieve the agile goals as effectively as possible, this is done through many different methods.

There are multiple agile aspects that I can incorporate into my prototyping, given GitHub has Kanban natively I will be using that agile method in my prototyping. In software Kanban are virtual cards that have, a request for code, a feature, a user story solution. I will have features that user suggest on my cards and prioritise and move them through the stages of development, testing, and final implementation. I could attempt to implement others such as XP however this is much more team focused just like other agile aspects like scrum. XP has something called pair programming where you program with someone looking over your shoulder to make sure the code is of high quality, and scrum is about meeting effectively to establish what needs to be done. I am coding alone (relatively) so the methods mentioned above are useless for me, for this task.

**Decomposition**

GUI

* Add a user (1-2 hours)
* Search for a movie (1-2 hours)
* Rate a movie (1-2 hours)
* Recommend (given the engine, 1-2 hours)

Storage

* Users (1-2 hours)
* Movies (1-2 hours)
* Ratings (training data) (2-4 hours)
* Neural network formatting (this depends on the engine, but I assume 1-2 hours for most engines)

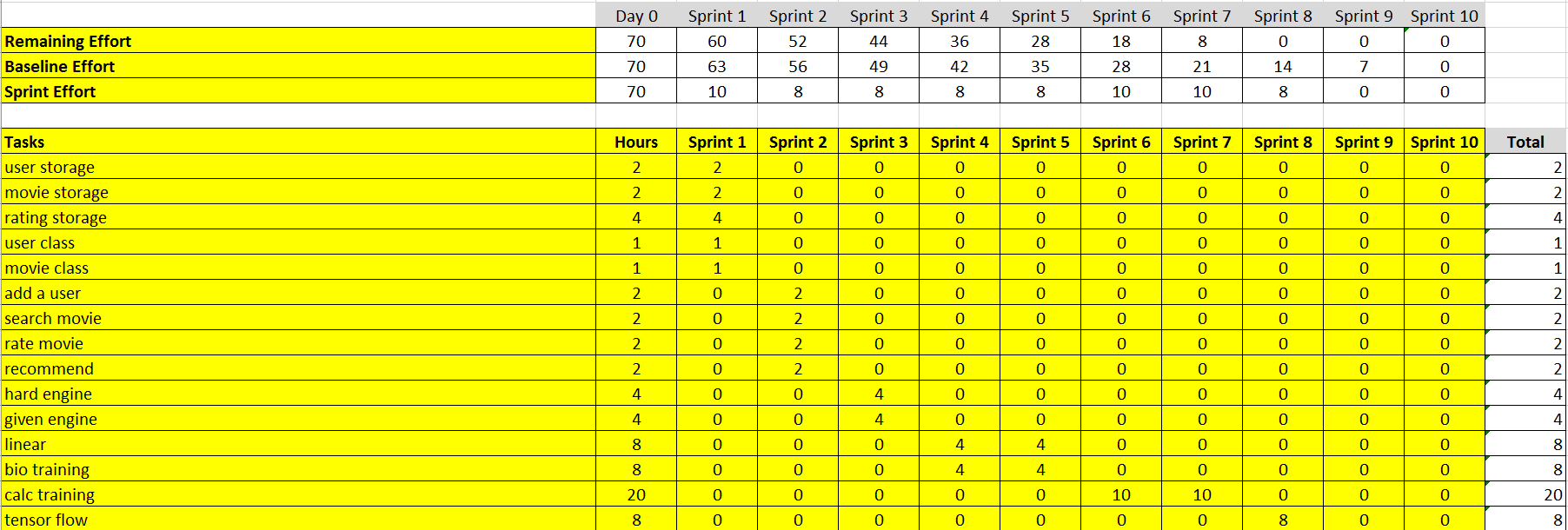
AI (these would progress with time)

* Given engine (2-4 hours, total guess)
* Hard coding (2-4 hours, total guess)
* Neural net from scratch (linear) (4-8 hours, total guess)
* Neural net from scratch (non-linear) (10-20 hours, total guess)
* Training with bio (4-8 hours, total guess)
* Training with backwards calc (10-20 hours, total guess)
* Tensor flow (4-8 hours, total guess)

Backend

* User class (1 hour)
* Movie class (1 hour)
* GUI functions (4-8 hours)

**Schedule**



I want to have some spare time at the end to give myself some room for error.

**Analogue prototype**

There are a few things that I want to try.

Having the inputs on the main menu and every time the confirm button is pressed for a given function it opens a window, or a frame.

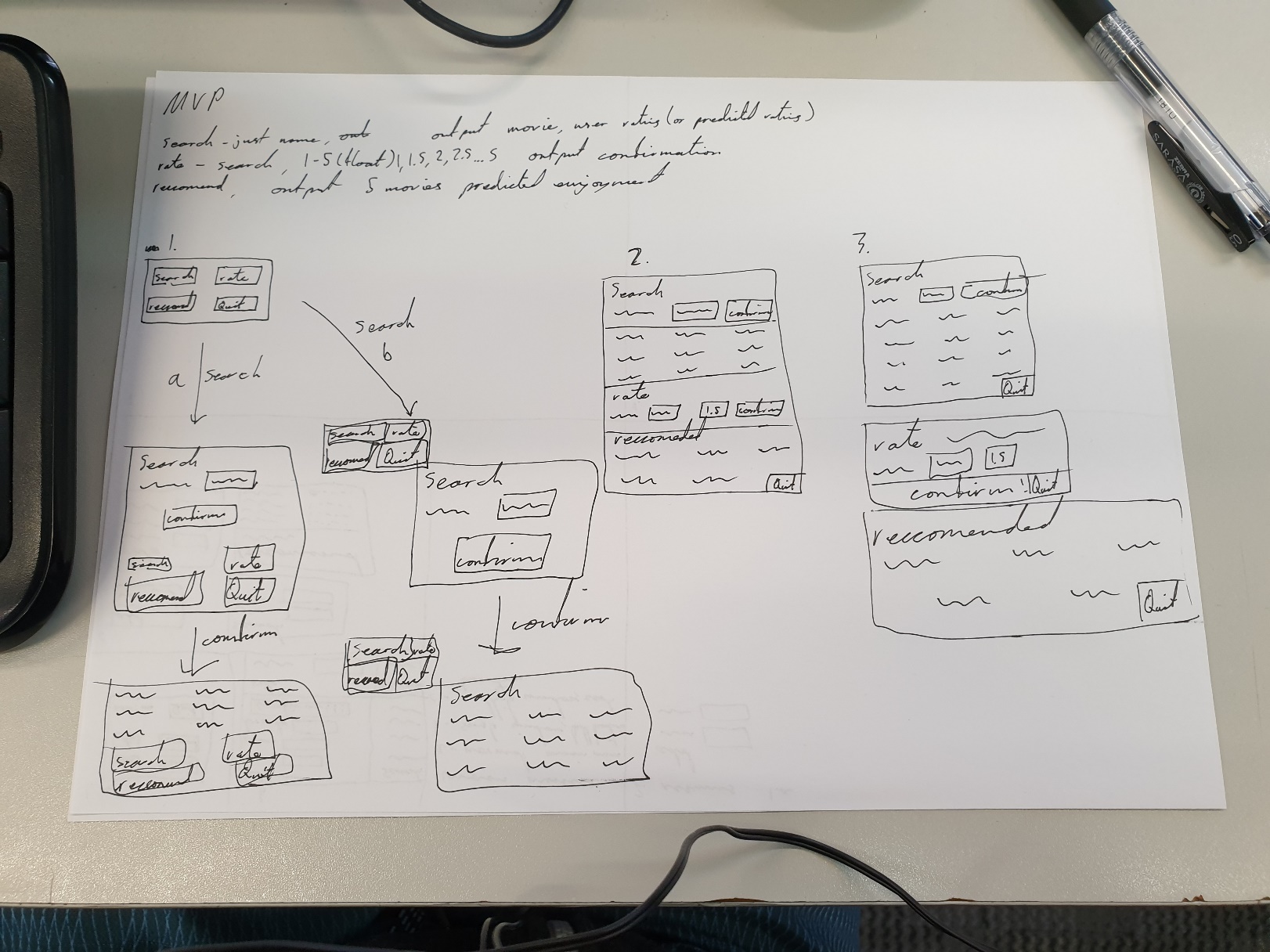
Having a menu of buttons that opens a window or a frame for the given function.

Multiple windows for each function

Anything user suggested

Should the search be a filter or name based search?

Should I log on as a user in the database?



Ashan’s feedback

1a. It’s simple, that it goes from the search and it stays in the same window, once you click it because it is the same window it is less confusing.

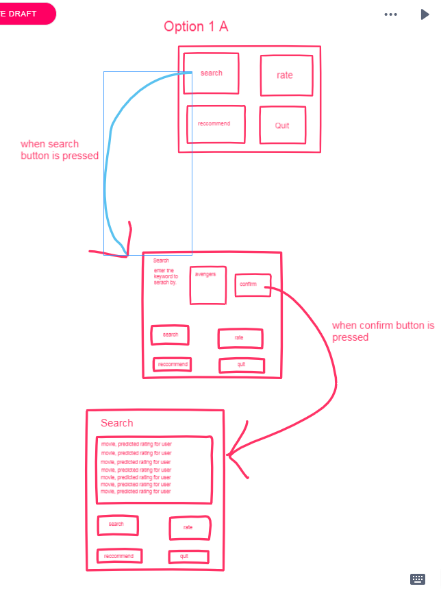
1b. Too complicated, doesn’t follow regular heuristic.

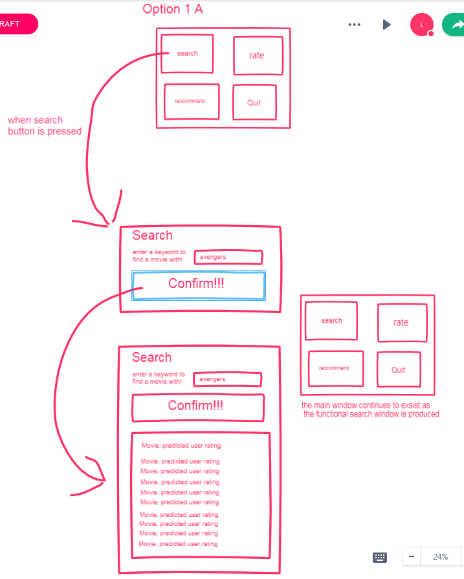
2. I feel like it could be cool and its giving a YouTube vibe, where you have a search bar and recommendation on the same place. You could take another hint from YouTube where the other functionality collapses after the search.

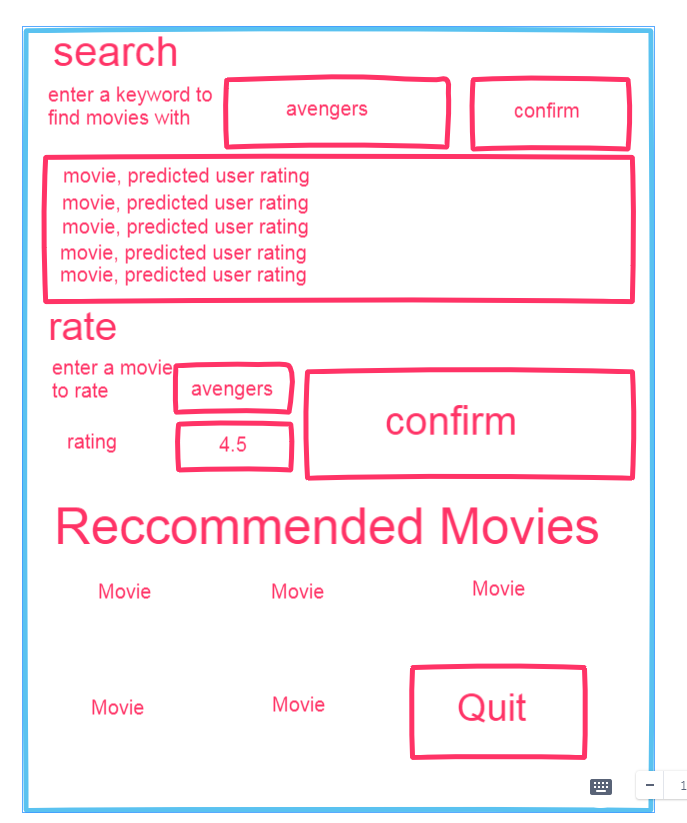
3. Absolutely no, it looks discostang.

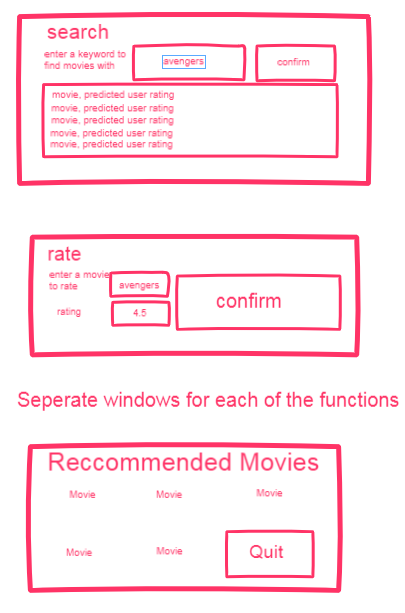
Mr Ny’s feedback

Make your storyboarding on a digital format, make it clean, professional and easy to follow without explanation.

1a.

1b.

 2.

3.

Mr Ny’s feedback

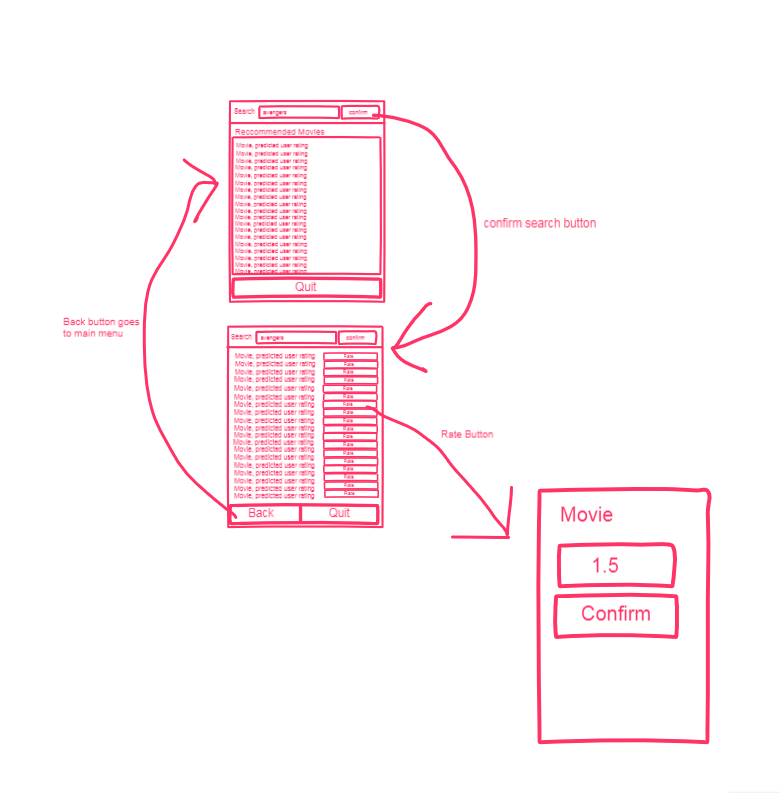
1a. no need for a rate as when movie is searched, you can rate it

1b. try to avoid multiple windows, it’s a pain in the ass

2. this is clearer, makes it easier for the user

3. try to avoid multiple frames

As both user have indicated a liking to 2. I will do the development suggested by Ashan. And the system where you can rate a searched movie.



Feedback: Ashan

Its aight, search bar with search being the confirm button therefore less clutter. Its sweet.

Tim Fahey

Just a few lines for everything, make some spacing just for less clutter, add some back buttons

So this looks mostly good, a few tweaks will be needed but they can be done later given the base structure seems to be liked in general.

**Programming**

The first task required is the user storage. Storing the users and their corresponding ratings. I will have to make the classes for users first.

In considering the user class I needed to consider whether a user should be able to change their ratings for a movie, for now I will not allow, but in later iterations it will be possible.

1. I took the cheap and easy tips from here <https://www.allaboutvision.com/cvs/irritated.htm> [↑](#footnote-ref-1)
2. <https://www.psychologytoday.com/intl/blog/you-illuminated/201010/why-your-brain-needs-water> [↑](#footnote-ref-2)