

# ComSci Project - MyTime

## Analysis Draft

Max Stupple

## Part I

# Analysis

### 1 Overview of the problem

Many students struggle to manage their time. The workload of A-levels alone is enough to make it difficult to balance between small pieces that are due in soon, and longer project that need a little bit of work here and there, let alone extra-curricular activities and sports that take up more time after school and on weekends. One solution is to manually plan how you will use all your free time, but this has two drawbacks:

- Many people struggle to estimate how long a task will take them and hence struggle to allocate a reasonable amount of time to each task
- This process takes time, a resource which has already been established to be finite and valuable

My stakeholder is Eli, a Year 12 Sixth Form student studying Maths, Further Maths, Physics and Chemistry. Eli struggles to fit in all his work around his various extra-curricular activities, so he needs an app that will not only help him keep track of what he needs to do, but timetable when to do each task and prioritise those which are most urgent.

### 2 Limitations of current system

To organise his tasks, Eli currently uses the Apple Reminders app on his iPhone. However, he finds this lacking for a number of reasons. Although the app helps him keep track of the tasks he needs complete and when they are due, it does not help him prioritise these tasks or inform him of the relationship between the amount of time he needs to do those tasks and the amount of time he has before they are due. He also doesn't find the app very engaging, as he lacks a sense of accomplishment after completing a task and ticking it off his list. Furthermore,

the app only provides the ability to sync between Apple devices, which he finds limiting as he cannot access his tasks on his Windows computer. Eli also finds the light theme of the app abhorrent, and desires a solution with a darker, more attractive colour scheme.

### 3 Proposed solution

My proposed solution is a program including the following features:

- Keep a list of the users tasks which contain a description of the task, a due date and time estimate
- Schedule tasks in user's free time according to a calendar and school/work schedule
- Record and track the actual time taken to complete a task, including breaks
- Provide feedback on the accuracy of the user's time estimates and productivity levels
- Adapt to the user's preference in terms of length of work sessions
- Display these tasks to the user in an organised manner using a Graphical User Interface

The program will be created as a web app using the Django web framework. The reason for creating it as a web app is so that it is available as widely as possible, as it will be possible to access it on any device with a modern web browser. I am using Django as this allows me to write the program in Python, the language I am most familiar with.

### 4 User group

My target users are students, in particular Sixth Form and University. As the program is targeted at individuals who struggle to manage their time and avoid procrastination, the program will need to be engaging and provide incentives for the user to complete tasks early rather than delaying them. Thus the user will avoid situations in which they find themselves with insufficient time to complete all their tasks before they are due.

The UI must also be simple and intuitive to use: there's no point in using an app to organise your time if you waste more time trying to get the app to work than doing the work you need to do.

## 5 Computational methods

This problem lends itself to a computational solution in particular due to the need for automation and interactivity to ensure engagement. One potential non-computational solution could be a physical calendar or to-do list, however this would not be able to provide the features required by my client. Such a solution could not automatically allocate time for the user to complete their tasks in, and could not remind the user of their tasks - it would require the user to check and plan the time for themselves.

### 5.1 Abstraction

Tasks will be stored as fields in a database consisting of columns and rows, but I will be displaying this in the form of a list of items with tags and attributes. Data will also be stored in a database but displayed in the form of graphs.

### 5.2 Reusability

There are a number of functions which my program will need to perform where it wouldn't make sense to write them myself from scratch, so I will use libraries that have already solved the problem. I will need to be able to get the current date and time, to associate with each task, and a SQL database to store my data in.

### 5.3 Visualisation

My program will need to be able to present data to the user in a way that is visually appealing and easy to understand. For example, data about the number of tasks completed can be presented on a histogram.

### 5.4 Concurrency

The program will need to be able to perform certain tasks in the background without interrupting the user, for example allocating time slots to tasks and analysing data to create graphs and provide feedback to the user.

### 5.5 Data Mining

Albeit on a small scale, my program will use the concept of data mining to analyse trends in the user's completion of tasks, such as how much time they spend and how accurate their time estimates are, in order to give feedback and help the user improve their efficiency.

### 5.6 Logic

The program will need to be able to intelligently allocate time slots to tasks based on the user's schedule and the time needed to complete the task. It will

also need to be able to adapt the user's habits and preferences regarding their work schedule.

## 6 Research

I identified a number of candidates for solutions to my stakeholder's problem. The candidate programs are:

- Forest
- Evernote
- Todoist
- Remember the Milk
- Ike
- Google Keep
- Trello

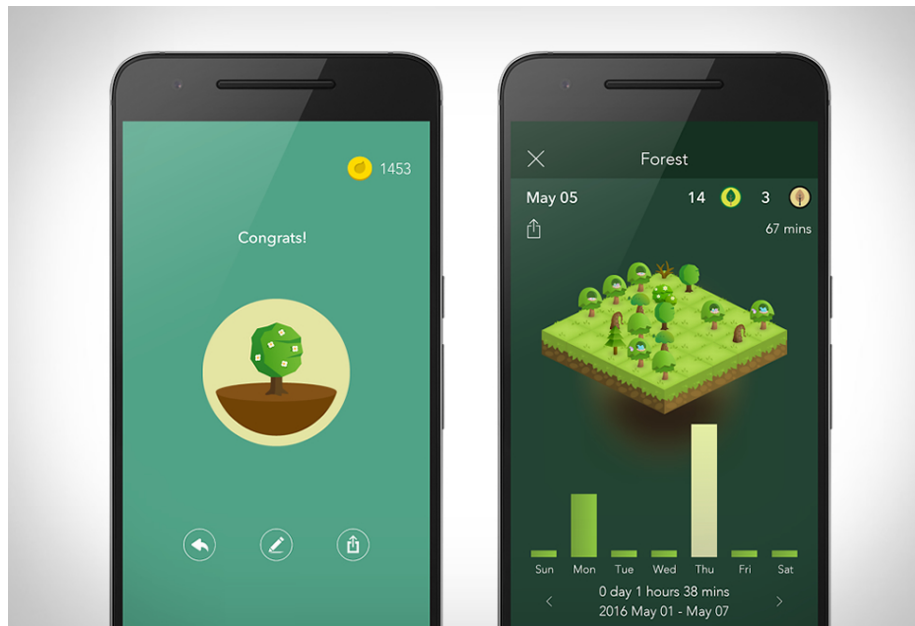


Figure 1: Forest

I think these programs all offer partial solutions to the problem, but none of them offer a solution to the exact problem my stakeholder has described.

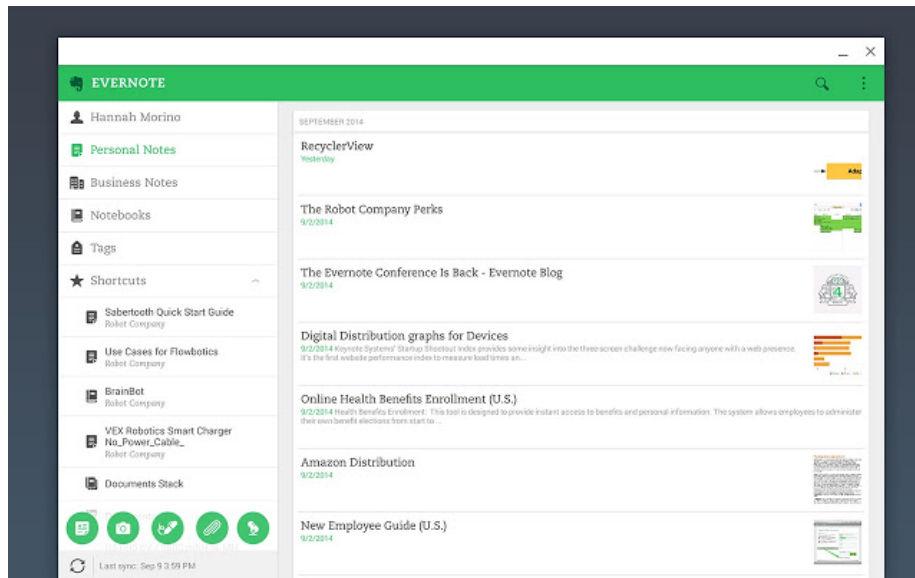


Figure 2: Evernote

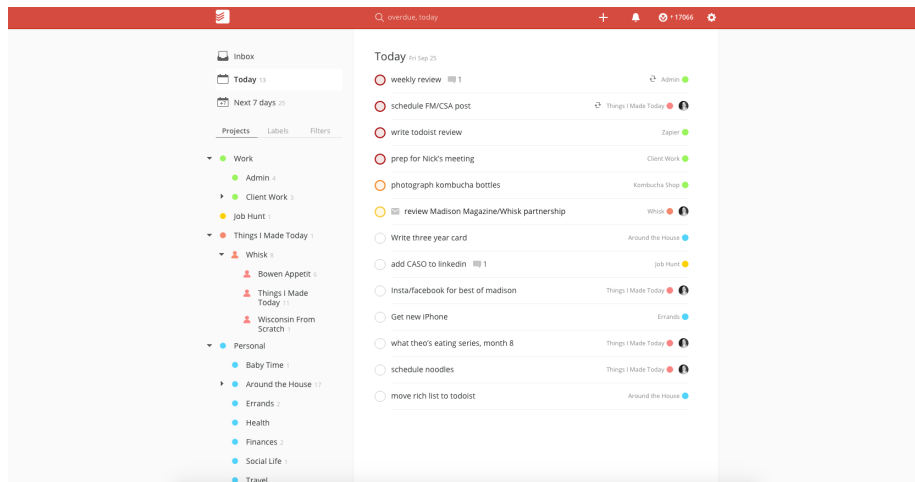


Figure 3: Todoist

Forest is excellent for helping you focus on a task, but can't keep track of your to-dos. Todoist offers excellent functionality for keeping track of and organising your tasks, but doesn't do anything with regard to helping you timetable everything that you need to do. Keep is better in this regard as it integrates with GCal to display tasks in your calendar, but can't allocate them those time slots automatically. Ike has a very appealing UI and a good system for organ-

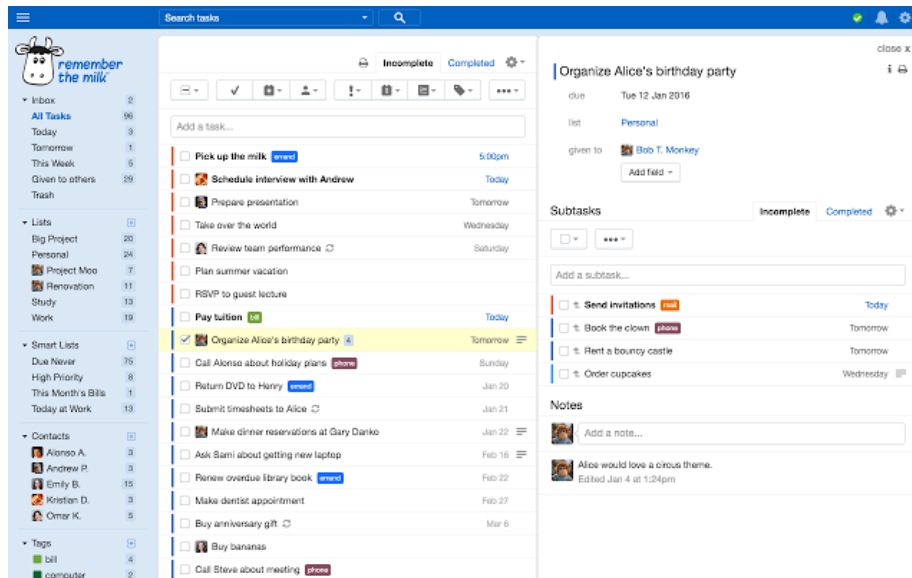


Figure 4: Remember The Milk

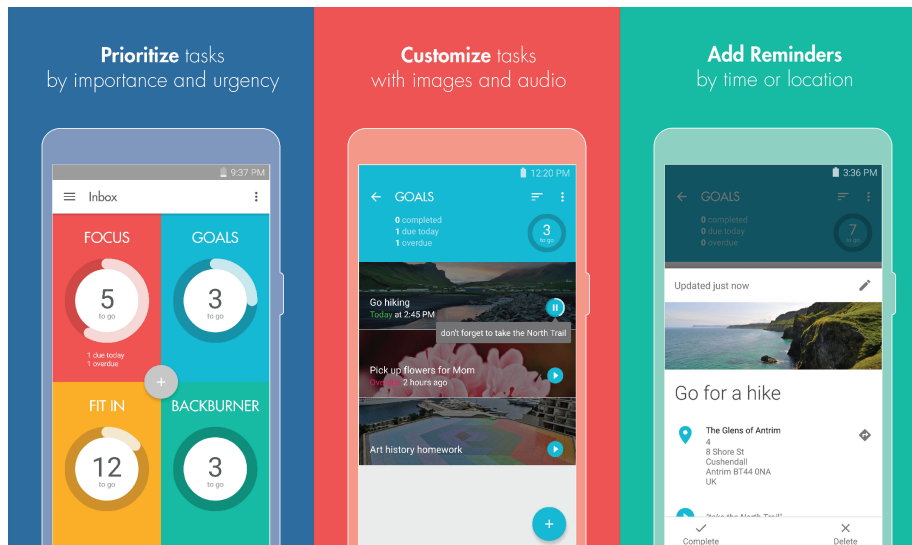


Figure 5: Ike

ising into four overarching categories, but you can't create your own categories like Todoist. Remember the Milk is probably the most intelligent of the programs, with a "Smart Add" feature that makes adding tasks very simple, and a powerful search for filtering through your tasks, in addition to integrating with

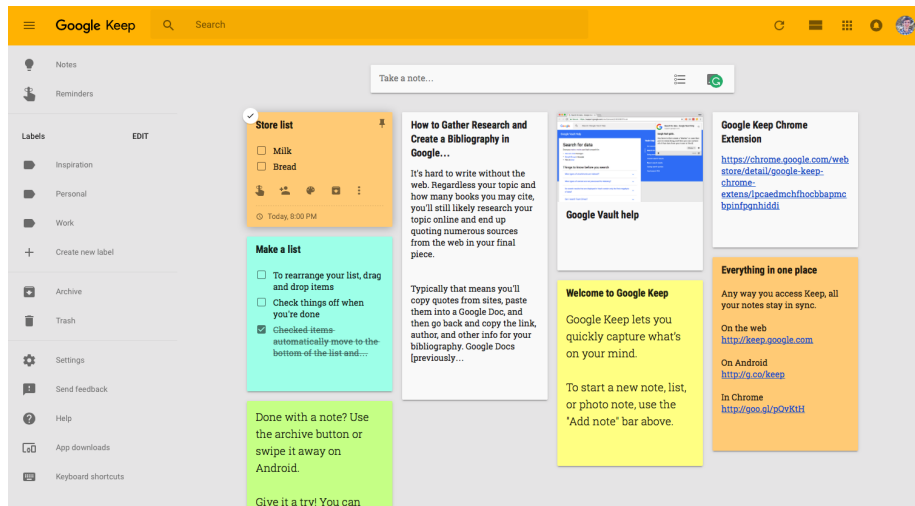


Figure 6: Google Keep

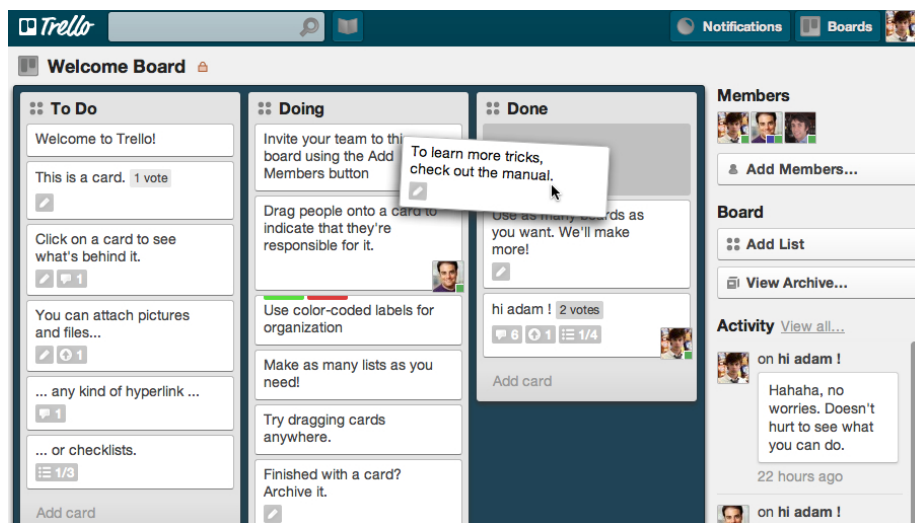


Figure 7: Trello

a number of other services such as email and social media for reminders, and cloud storage services for adding attachments to tasks, however this is probably beyond the scope of the problem I'm trying to solve. Evernote is in my opinion the least effective of these programs, as it is mainly focused on note taking, with reminders as a side-feature. Trello offers the most features oriented towards time management and prioritising tasks, but is more focused around team collaboration on big projects than individual to-do management.

I showed the candidates to my stakeholder to get his opinion and he gave me the following comments:

Forest: “This is my favourite. The UI is excellent and the metaphor of growing trees is very appealing. Out of all the programs this does the best job of helping me manage my time, however it’s unfortunate that it doesn’t include integrated task management. I like that you can see your past progress as this is very motivational, and it stops the timer if you leave the app, which helps you avoid idly switching to Facebook or Twitter for a ‘quick check’.”

Evernote: “Good for note taking, but that’s not really what I’m looking for. It has too many extraneous features, which are unnecessary and make it feel bloated - I want a more streamlined experience. I also dislike the subscription model.”

Todoist: “Great for managing tasks, with graphs and data to track your statistics. I love the categorisation and colour coding for different tasks, and being able to give them different levels of priority. I also like that you can export your tasks to your calendar. The lack of a dark mode harms the UX.”

Remember the Milk: “There’s too much task segregation which makes the UI confusing. It also has a weird notes system. Not a fan.”

Ike: “The idea behind it is admirable, but ultimately the categories feel a bit arbitrary, and that’s made worse by the lack of an ‘all tasks’ view. The UI is very clean however, and the animations are really nice.”

Keep: “It’s good for lists, but otherwise nothing special.”

Trello: “Good for project development, but not well-suited to personal task management.”

He also commented in general that he liked the ability to sync tasks between devices, and a feature which he wanted but none of the programs offered was the ability to have “subtasks” nested inside other tasks.

From this I have assembled the following list of features which my program will need:

- Main view displays all uncompleted tasks and recently completed but not-deleted tasks
- Archive containing completed tasks, and allows tasks to be unmarked as complete
- Tasks grouped in categories, can be colour coded
- Tasks can be filtered by category



- Ordered by time needed or due date
- Tasks can be marked as done or archived
- New tasks can be added, with a brief title, optional additional notes, an estimate of time needed and a due date
- Graphs showing number of tasks completed, amount of time taken, and whether tasks were completed on time
- View data for day, week, month, year and all time
- Show upcoming tasks in their automatically allocated time slots
- User can enter the schedule and other commitments that the program will schedule tasks around
- User can manually change the allocated time slot for tasks
- The program will give a warning if there is not enough free time to complete a given task before it's due date
- Current task displayed at top of screen
- Time spent working and time to next break
- Buttons to manually pause timer and take a break or mark task as done
- Graphic showing a town/city building up over time as you work

I showed this list to Eli, and he added that tasks should be given a priority level, so that high priority tasks can be scheduled before low priority ones. He also elaborated on the city-building mechanic, resulting in the following:

- The city builds over time as you work
- Taking a break which has been allocated by the app simply pauses development
- Taking an unallocated break sets the development back - perhaps there is a level system and you can be set back one or two levels
- If you quit a task before you finished - and taking an excessively long break automatically quits - the city is destroyed
- If you take too long to complete a task, development is slowed down
- When you finish a task, you can either stop, which doesn't destroy your city but it degrades over time, or go straight to the next task, in which case progress continues
- If you finish a task early and go straight on to another task, your city gets a boost

Eli said he “agrees with all of this” and called it “good design”. He also emphasised his desire to access his tasks across different devices. I have concluded that the best way to facilitate this would be to build the program as a web app.

Eli also suggested that there were psychological benefits to offering the user a choice in what task they do. Studies have shown that individuals are more motivated to complete a task which they have chosen to do from a set of options, rather than only one. Therefore I will endeavour to implement a system which, rather than forcing, or heavily encouraging, the user to complete one particular task in a certain time slot, will instead give them the option to choose between tasks with similar levels of priority.

## 7 System requirements

As the program will be web-based, it will require a system capable of running a modern internet browser, such as Firefox. The system requirements for Firefox 66.0 are as follows:

### Windows

#### Operating Systems (32-bit and 64-bit)

- Windows 7
- Windows 8
- Windows 10

#### Recommended Hardware

- Pentium 4 or newer processor that supports SSE2
- 512MB of RAM / 2GB of RAM for the 64-bit version
- 200MB of hard drive space

### Mac

#### Operating Systems

- macOS 10.9
- macOS 10.10
- macOS 10.11
- macOS 10.12
- macOS 10.13
- macOS 10.14

## Recommended Hardware

- Macintosh computer with an Intel x86 processor
- 512 MB of RAM
- 200 MB hard drive space

## GNU/Linux

### Software Requirements

*Please note that GNU/Linux distributors may provide packages for your distribution which have different requirements.*

- Firefox will not run at all without the following libraries or packages:
  - GTK+ 3.4 or higher
  - GLib 2.22 or higher
  - Pango 1.22 or higher
  - X.Org 1.0 or higher (1.7 or higher is recommended)
  - libstdc++ 4.6.1 or higher
- For optimal functionality, we recommend the following libraries or packages:
  - NetworkManager 0.7 or higher
  - DBus 1.0 or higher
  - GNOME 2.16 or higher
  - PulseAudio

Any system which meets these requirements will be able to run the program.

## 8 Success Criteria

### 8.1 General objectives

To create a program which stores tasks and arranges them around the user's schedule. The program should engage the user through the use of a game-like progression system and help the user complete their tasks in a timely manner through the use of reminders.

## 8.2 Specific objectives

The program should:

- Store a list of the user's tasks
- Store the due date, priority, expected time needed, and other information about each task
- Add and remove tasks from the list
- Allow the user to group tasks into categories of their choosing, and colour code categories
- Record the successful completion of each task, time taken, and number of breaks taken and display this information to the user in a useful manner
- Schedule time for the user to complete their tasks, according to the user's schedule, task due date, task priority, and the user's preferences
- Display the tasks in their allocated time slots in a calendar view, and allow the user to manually alter time allocations
- Have a focus mode, which helps the user concentrate on the task at hand, and incentivise the user to complete the task in a timely manner without procrastination using game-like aspects
- Be available on multiple platforms and devices
- Sync tasks between devices
- Remind the user of upcoming tasks