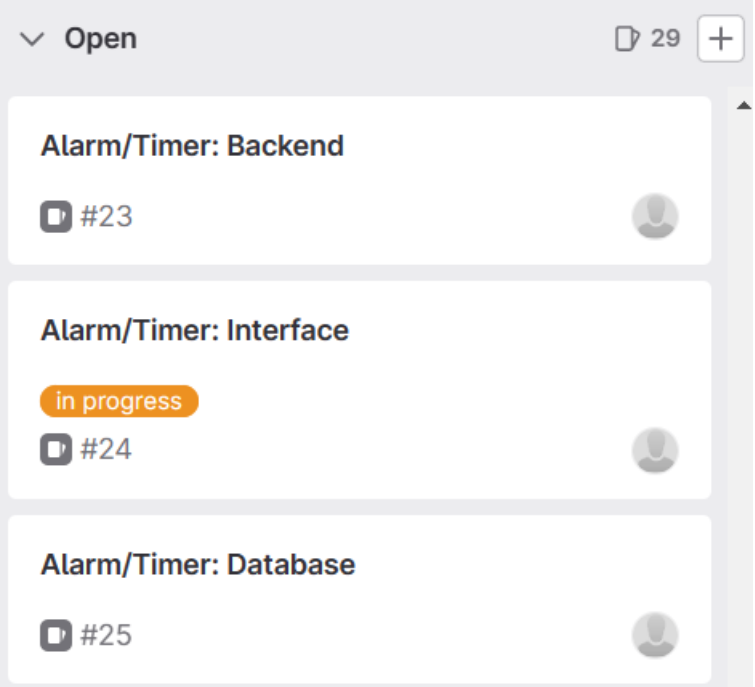
S2 Submission: Matthew Goulding 2330080

Agile Estimation of Feature Cards



Backend: 12 hours

The backend implementation will be making the alarm and timers functional. I need to code it so that the user can create timers and alarms and program buttons to start and pause them separately. I have estimated that this will take 12 hours to implement.

Interface: 8 hours

The appearance of the page where timers and alarms are created and visually displays each one with the time remaining/accumulated with the task name and buttons to interact with each alarm/timer. This is a slightly shorter task and I've estimated it will take 8 hours.

Database: 10 hours

Coding the ability to read from and write to the database to allow for the collection of statistics which will be used elsewhere. I have estimated that implementation will take 10 hours

DevOps Tech Report

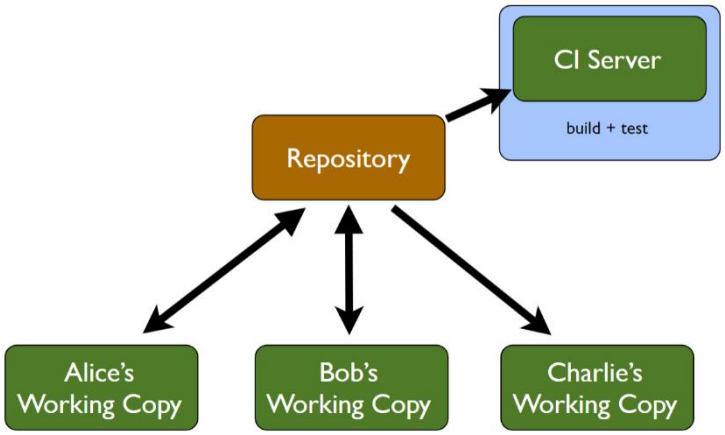
DevOps is a software development methodology that emphasizes collaboration and communication between software developers and IT operations professionals. It seeks to streamline the software development and deployment process by automating and integrating the development and operations teams' workflows and tools.

DevOps promotes a culture of continuous integration, delivery, and deployment, which means that changes to software code are continuously tested and released into production. This approach enables teams to respond quickly to changes in the market or business requirements, reduce time-to-market, and improve the overall quality of the software product.

In DevOps, a pipeline is a series of automated steps or stages that software code must pass through, from development to deployment. The pipeline includes the steps required to build, test, and deploy software in a consistent, repeatable manner.

Typically, a pipeline starts with the development team writing code, which is then committed to a version control system. From there, the pipeline uses tools like Continuous Integration (CI) and Continuous Delivery (CD) to automatically build, test, and deploy the code to various environments such as testing, staging, and production.

In our project we are using GitHub for a Continuous Integration server which allows the programmers to each have a working copy of the code from the repository that they can update separately. Continuous integration servers reduce the integration problems significantly by running automated checks and tests every time changes are committed to the codebase. This ensures that any integration issues are identified and resolved as soon as possible. Continuous integration servers also reduce the time taken to release new versions of software as they automate the build and deployment process.



We are using an Amazon EC2 virtual machine in our project. Using a virtual machine for programming in a team allows developers to collaborate on a project without running into compatibility issues. By running a virtual machine, each developer can have their own operating system and development environment that is tailored to their needs. This makes it easier to test and troubleshoot code while ensuring that everyone is working on the same platform. Additionally, virtual machines are more secure and can be used to store confidential information. This makes it easier for teams to collaborate without worrying about data breaches.

Development prep: tech stack/CI

```
PS C:\Users\matth\git\team23-22> git add .\BranchTest.txt
PS C:\Users\matth\git\team23-22> git commit -m "branch test"
[MVP_AlarmTimer 327a1fd] branch test
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 BranchTest.txt
PS C:\Users\matth\git\team23-22> git pull
From git.cs.bham.ac.uk:team-projects-2022-23/team23-22
 * [new branch]      MVP_Antiprocrastination -> origin/MVP_Antiprocrastination
Already up to date.
PS C:\Users\matth\git\team23-22> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 275 bytes | 275.00 KiB/s, done.
Total 3 (delta 1), reused 1 (delta 0), pack-reused 0
remote:
remote: To create a merge request for MVP_AlarmTimer, visit:
remote:   https://git.cs.bham.ac.uk/team-projects-2022-23/team23-22/-/merge_requests/new?merge_request%5Bsource_branch%5
D=MVP_AlarmTimer
remote:
To git.cs.bham.ac.uk:team-projects-2022-23/team23-22.git
   0934232..327a1fd  MVP_AlarmTimer -> MVP_AlarmTimer
PS C:\Users\matth\git\team23-22>
```