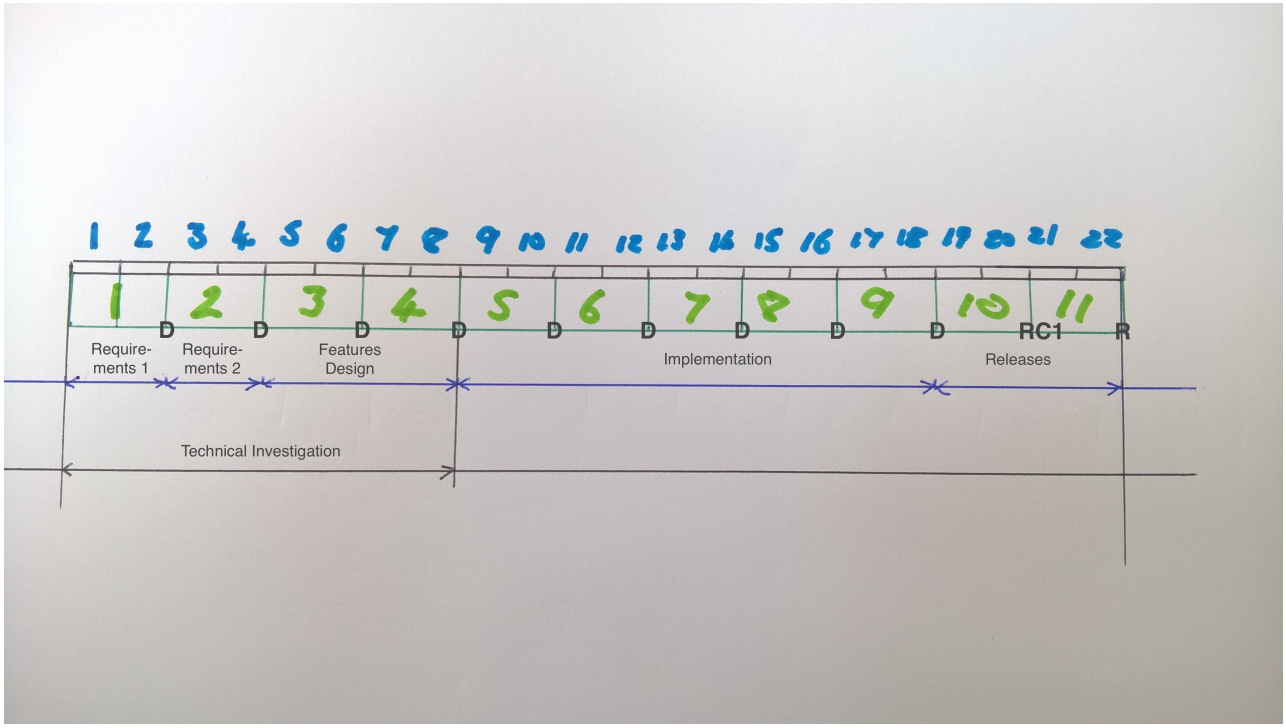
Your Project

Welcome to your Digital Labs Live Project!

Projects run from 1st Week in October, to last week in March, where there is a showcase event.
Project Time: 22 weeks
Team Size: 4-6 people

That's a long time, and alot to get organised. If you split the time up into project phases, it's much more managable. If you deliver regularly, you get plenty of feedback, and stay motivated.

Here's the Live Project timeline:



We've split it up into weeks, and sprints. We've made some project phases to indicate the activities which should be going on. We do this ourselves on the projects we do. It works really well, with a motivated team and enough people for the job.
Working on this project, you'll need to work to this timeline.

The rest of this article is a guide on what to do and expect as you work on this project.

Your Effort. How much?

Figure out how much effort your team can commit to your project. You can use this to imagine what it is possible to achieve, when you start to work out what needs to be done.

Personal Effort

Throughout this guide, we've set a standard for effort: it will effect what you can achieve. Numbers below are a guess at hours-per-person-per-week. They assume working through the entire 22 week period of the project.

Minimum: 5 hours per person per week You'll get something out of the project, but you'll have to cut out alot of functionality. Make sure that you're up-front about this in your requirements sessions, so you can plan a minimal implmentation. Do spend enough time in the technical investigations, so you don't get a nasty surprise later on!
Expected: 10 hours per person per week With some good design and technical investigation, you will be able to produce a result which does the job on all counts. Be very aware that the tasks we have set you concentrate on results. There's alot you can do by using tools and frameworks, that take you away from writing boiler-plate code, and concentrate on what's important.
Stretch: 15 hours per person per week You're ambitious, and really want to use this project to learn some amazing skills. Be careful though! It's really easy to waste alot of time on a cool feature at the expense of one which is essential. Stick to the plan, and make the extra effort pay off for you.

Team Effort

We've taken the values above, and added them together to give you an effort budget for the project. This represents the total amount of time that the team will have to complete its work. For a comparision, we've shown you the effort budgets for both a 4 person team, and a 6 person team.

Total Team Effort, over the whole project

4 person team effort (hours):

Minimum	Expected	Stretch
440	880	1320

6 person team effort (hours):

Minimum	Expected	Stretch
660	1320	1980

In the project timeline chart above, the project is split up into 'sprints' of effort. To help you work out how much you can acheive in a sprint, here are the numbers:

Total Team Effort, over a single sprint

4 person team effort (hours):

Minimum	Expected	Stretch
40	80	120

6 person team effort (hours):

Minimum	Expected	Stretch
60	120	180

Phases, Sprints and Deliverables

Here's some explanation of the activities you should be doing in the various phases of your project, to make sure you get a good result.

The requirements phases are important: this is where we can get an understanding of what you feel you can do over the project, and what you can deliver. Be frank about the effort you can put in, and realistic about the product that you think you'll get out.

The design phase is where you are confident you know what you want to deliver, and now want to make a plan. You can't make a plan without a basic design to work to; This is why we have a technical investigation running right from day one; to work out what is possible, and how long things take to do. Your technical investigation and your requirements should inform your design.

When you have a design, it's time to plan the implementation phase; take an informed guess at the effort it will take. Do this by breaking down your design into components, and working out the tasks it might take to create them. Assign effort to each task. The implementation phase is 5 sprints (10 weeks) long. If your guess doesn't fit into this timescale, you may need to cut out some features, to make it fit.

Deliverables occur at the end of every sprint, right through the project. Make sure you deliver them. In exchange, we will give you feedback and support! Where the deliverable is a document, make it sparse and clear. Don't waste time on formatting (Use markdown). We just want to know what you're doing and how we can help. It's the product which is important!

Finally, the release phase, is a time for tidying up, and making sure things are stable, before the big launch of your product.

Here are the phases in a bit more detail:

Technical Investigation

Object: Understand the problem and how a solution will be implemented.
This effort runs over 4 sprints, running in parallel with requirements capture. The whole team is involved. This is an investigation of all the things needed to implement the project. How they will work, what pitfalls there might be. Build up a good set of example code, links to resources, etc.
Deliverable: Evidence.
... in the form of source code and working demonstrations, backing the requirements capture.

Requirements 1

Object: Understand the client's needs.
Start with a requirements meeting with your client. Nominate a Project Manager, and Technical Lead. Gather requirements from the client and feed into the technical investigation. Project Manager and Technical Lead are responsible for interpreting the clients needs, and communicating them effectively.
Deliverable Document. First draft Requirements. This document is based on the first bits of the technical investigation and focusses on what might be possible in the project.

Requirements 2

Object: negotiate what is possible.
Use the technical investigation to figure out what will REALLY be possible in the implementation time available (5 sprints).
Have a second requirements meeting half-way through the sprint to comunicate with the client the technical findings, and negotiate the final requirements.
Write-up the requirements.
Deliverable: Document. The team's understanding of what the project will produce.

Provide a description of the product to be delivered:

- break the product down into features (wireframes / supporting documents as necessary)
- describe the beheviour of the product
- describe what technologies will be used.

Provide a first-guess estimate:

- how much effort (in developer-hours) it will take to do each feature.
- how much effort (in developer-hours) it will take to do the whole project. That's features plus testing, integration, and contingency for when stuff goes wrong.
- seperate the features into 'essential' and nice-to-have.
- define the minimum viable feature-set.

Features Design 1

Object: Produce an underlying design on which to develop the features. We're looking for a really good understanding of the requirements, with a design and some proof that it can be done, informed by ongoing technical investigation.
Deliverable: Document. Initial Design. Seperation of features into components and behaviors. Look where you can find commonalities (Don't Repeat Yourself).

Features Design 2

Object: Produce a plan.

- Use the technical investigation to inform your refined design.
- Work out the tasks required to construct and integrate each one of the components into the product.
- Implicit in the task is repository / code maintenance and unit testing.
- Estimate how long each task should take in hours, by talking to the developers.
- Check that the tasks for the essential features fit into the Implementation time (5 sprints)
- Design compromises if the effort for the implementation of the features is too much.

For reference: here's how much effort goes into the 10 week implementation:

4 person team effort hours:

Minimum	Expected	Stretch
200	400	600

6 person team effort hours:

Minimum	Expected	Stretch
300	600	900

Deliverable: Document.

- Refined design: Features, Components, Tasks.
- Project plan: scheduling of tasks into 5 sprints of effort.

Sprints

Object: complete tasks, make a demo.

- Design, code, test, check-in. Repeat. Test the overall product.
- Communicate. Meet **every day** for 10 minutes to tell each other what you worked on, what you're going to work on, and what problems you had. BE KIND. Dont blame!

Deliverable:

- a demoable 'product'.
- a revised plan. (NOTHING goes to plan! All you can do is revise your understanding and figure out what to do next)

Feature Freeze

At the end of sprint 9 development of new features is stopped. Concentrate now on release. Think about the release process, testing, tweeking, stability.

Release candidate

Made available for general testing

Release

Final release for showcase.

Communicating

We just mentioned communication in sprints, but we thought it so important, it's got a section to itsself.

In your team

One of the first things you should do is appoint two very important roles:

- project manager
- technical lead

These two roles are no more or less important than the other members of the team. No dictators here. The roles are for communication.
The **project manager** gets an understanding of what is happening in the project, and communicates it to everyone within the project, and external to the project.
The **technical lead** is responsible for collating the technical investigations into an overall design, which is OK'd by everyone in the the team. They must communicate the ideas behind that design to the rest of the team, and communicate the technical ideas of the developers to the project manager.

Stand-ups

During the sprint phases, it's really important to organise who is doing what. The best way of doing this we've found, so far, is to hold a 'stand-up' meeting at the start of every session you are together on the project.
During the design phases, you created a set of features for your project, and a design. You made a plan, and worked out what tasks were needed to build what components, and in what order.
The stand-up is where you review project progress, and assign tasks to people, to build that design.
The project manager organises the stand-up.
Each team member in-turn says what they did yesterday, what they intend to do today and any problems they are having.
If anyone is close to finishing a task, they can nominate themselves for the next task, or have one assigned by the rest of the team, via the project manager and technical lead.
The stand-up is short; the name comes from people standing-up to have the meeting (if your feet get tired, you've been there too long. Wrap it up!)

With your client

That's us. DigitalLabs has lots of projects running this year. And it's really important to communicate effectively. That's why we've introduced Alice.
Alice is your gateway to DigitalLabs, for deliverables and support.
You will need to set yourselves up as a team in order to do this. You can find out how to do that in the Alice Wiki.
Most important, though: all communication with DigitalLabs needs to be via your project manager, and Alice. That makes it easy 😃

Thanks for reading. Good Luck!