DuckDebug

White Paper

Team Golf

Contents

Introduction

In recent years, the interest in programming has increased significantly. This has resulted in the birth of websites that are specifically tailored to the world of programming such as online code hosting sites, for example, GitHub, Bitbucket and SourceForge as well as code collaborative sites that provide real time collaboration such as Codebunk, Cloud 9 and MadEye.io. These sites allow the increase of productivity among groups of aspiring programmers by providing them with a workspace that can be accessed anywhere and can be shared with anyone in the world. This also allows the choice to ask for help on certain aspects of programming such as how to program a function or simply finding an error in a piece of code.

The Problem

To the inexperienced programmer, discovering errors is a common issue during the development of software, thereby making the process a frustrating experience due to having to spend valuable development time on solving the problem before the developmental phase can continue thereby blocking the entire software development process.

Even though there are many sites that can ease this process problem by providing a social and teamwork aspect to programming, users cannot be sure as to who they are communicating with and even if they do provide an answer, there may be a better solution to be found elsewhere.

Even though the programs themselves prove to be a valuable and easy to use asset once completed, their development can be exhaustive and highly stressful once problems are discovered. But how does one simply find a definitive answer to their solution, with the knowledge that they have full faith in who they are conversing with and what they are trying to provide.

How Can We Solve This?

DuckDebug is a new revolutionary idea to solve this gap in the online programming world. It is an online ruby based interface that will be specifically designed to allow users the ability to upload broken pieces of varying size of code that can be solved for a fee by a team of registered and certified programmers.

The DuckDebug interface provides the customer with a guarantee that the individuals whom they have placed their code with, are reliable and can be trusted to find the solution to their problem. This is achieved through the requirement for the programmers themselves to prove their knowledge upon registering their programmer account with the application. This prevents those who would be seeking to exploit the software for their own financial gain, from being enrolled.

To provide the best customer satisfaction during the problem-solving process, a communication feature is implemented to allow both the customer and programmers to interact with one another during the process. Customers can use this to check up on the progress of their solution to ensure that they are getting the best value for their money.

As well as providing features for the customers, the programmers themselves can benefit from using DuckDebug. Firstly, they are able to profit from using their valuable knowledge and skills to solve 'code bounties' that customers have uploaded to the system.

Secondly, programmers can browse a list of these bounties to decide which problems they would like to solve thereby enabling them to focus their specialised knowledge on their preferred areas of programming.

Features

To provide the best customer satisfaction when using the DuckDebug application. A number of features are provided to enable users to achieve the best experience possible. These include:

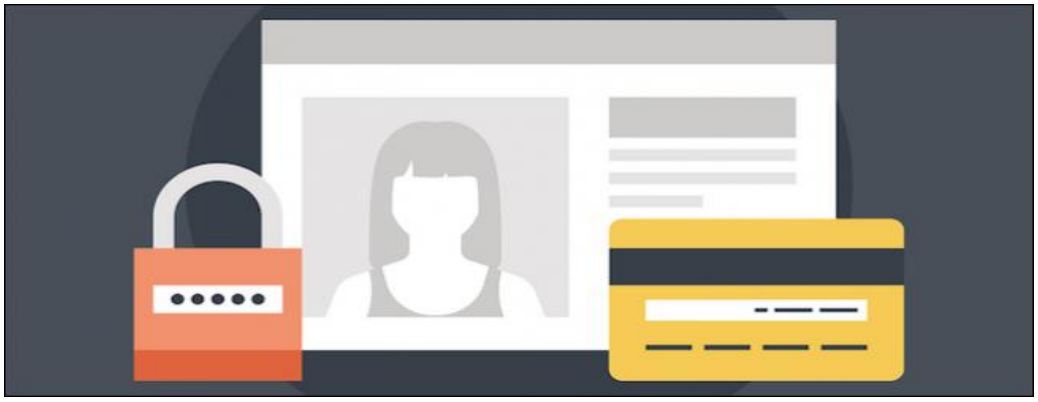
- Code bounties
- Bounty claim system
- Secure payment system
- One to one interaction between programmers and users
- Online hosting of code
- Separate user and programmer account systems

The DuckDebug system is designed around the aspect of allowing both programmers and customers guaranteed reliable, easy and fluid user interaction every time they use the system by using seperate programmer and customer accounts. This gives the system a dynamic approach to problem solving by enabling both sets of users with the tools required to co-operatively work together to reach the goal of solving the bounty.

The main feature of DuckDebug is the bounty system and is the heart of the operational running of the application. It allows registered users to submit their pieces of problem code known as 'bounties' in exchange for a solution. Registered programmers can then browse a list of these bounties and can use filters to sort them into different lists depending on the skills that they have.

When solved, the user who owns the bounty is required to pay a small fee to both DuckDebug and the programmer who solved the bounty itself using the system's secure payment feature. This payment is then transferred directly to both the system (20%) and the programmer, securely as per the Payment Card Industry Data Security Standards (PCI DSS).

The accounts feature of the system allows both users and programmers to perform a variety of actions with user accounts including features such as bounty submissions, payment options and account editing.



Our Business Case

In order for DuckDebug to generate revenue, a 20% share of every bounty shall be taken, this is in order to produce profit as well as to upkeep the price of running costs. Customers shall register with the system and upload their code with an attached bounty depending on the difficulty of the issue with bounties starting at as little as £1. Once a bounty has been solved, it is held in the system until customers have paid the total price of the bounty in order to receive the solved piece of code.

At the present time, there is very little competition in the open-source bounty market. This is due to the technology being relatively new as interest in programming is increasing in today's world. One such site, Bountysource, is based on this system, however, their focus is more towards the addition of functions to existing software, rather than solving bugs in code. This is where DuckDebug comes into its own, by becoming a platform that can serve a large number of customers with varying degrees of coding issues, in one centralised location, that provides an efficient and easy to use way of collaboratively solving bugs in code.

Marketing

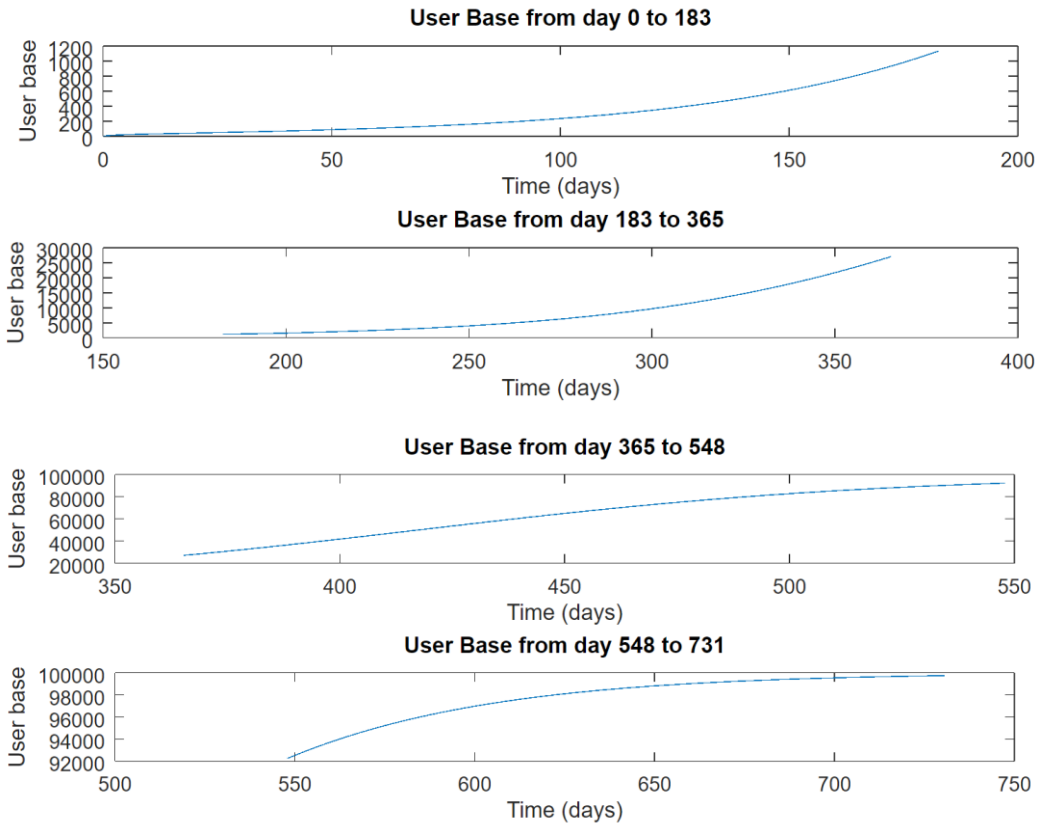
DuckDebug's marketing campaigns will involve marketing agents visiting online programming companies. More specifically, sites that are popular with the general public, such as Google through the use of google ads, as well as those that are involved with the teaching and collaboration of programming related tasks and projects such as GitHub and Codecademy. These campaigns include implementations that sites can make such as online adverts, that provide site visitors with a link that once clicked, shall send them directly to the DuckDebug application.

Financial Plan

In order to make a profit, the following is an example of what can be achieved with this software for a bounty with moderate difficulty:

The bounty price for this level of difficulty is estimated to be around £25 with DuckDebug claiming a 20% share of this amount equating to £5. Once DuckDebug has reached its borderline revenue before a profit is made, there is estimated to be an average of 175 bounties completed a day making a total of 192,000 a year equating to around £960,000 over a period of three years.

To answer whether this is achievable, we have modelled our growth rate based on the rate of similar companies:



Assuming this to be an accurate representation given our parameters, our goal of £960,000 would certainly be achievable.

Technical Details

Duckdebug.me is a ruby based, web orientated piece of software which is designed to allow users (customers) to post chunks of code ranging from all sizes and offer up a bounty to the anyone(developers) willing to fix it.

Technologies

Ruby on rails was the technology we decided to use for this project as with all the tools available it makes it easy to build complex websites in less time. Ruby on rails has many liberties (gems) available and has a large user community. Ruby is modern language will be used for generations to come with plenty of developers, as it known as one the easier programming languages to learn.

Database management systems

For our database management system this will be done in PostgreSQL. Postgres is somewhat lenient in licensing compared to other DB management systems like MySQL. Has a large vibrant community of PostgreSQL professionals and enthusiasts that can draw upon for support. It is very reliable and stable [1]"it is extremely common for companies to report that PostgreSQL has never, ever crashed for them in several years of high activity operation. Not even once." It supports all platforms and it is windows native. There are many high-quality GUI Tools available to suit the needs of all kinds of user from open source developers to commercial providers.

The system will be split down into two main Customer user types, they will have similar admin right but on the developers side the will be able edit code on in real time on the site or download a dump and edit on the personal machine then re-upload it to the website

Product requirements

We will host the website on a (server) the system will need to allow both set of users to be able:

- Login and Sign up. Create a profile and set up billing information
- Upload Problems / Projects / Code
- Offer bounties on resolving their problems
- Upload solutions to problems
- View and filter available bounties based on project languages and technologies specific to the developer's skill-set
- Input payment and cash-out methods
- Secure PCI DSS compliant payment methods
- Code upload of any size (free within reasonable limits - 100MB or more for a fee)
- Email notification when bounty was accepted or completed
- 95+% Service availability

Non-Functional

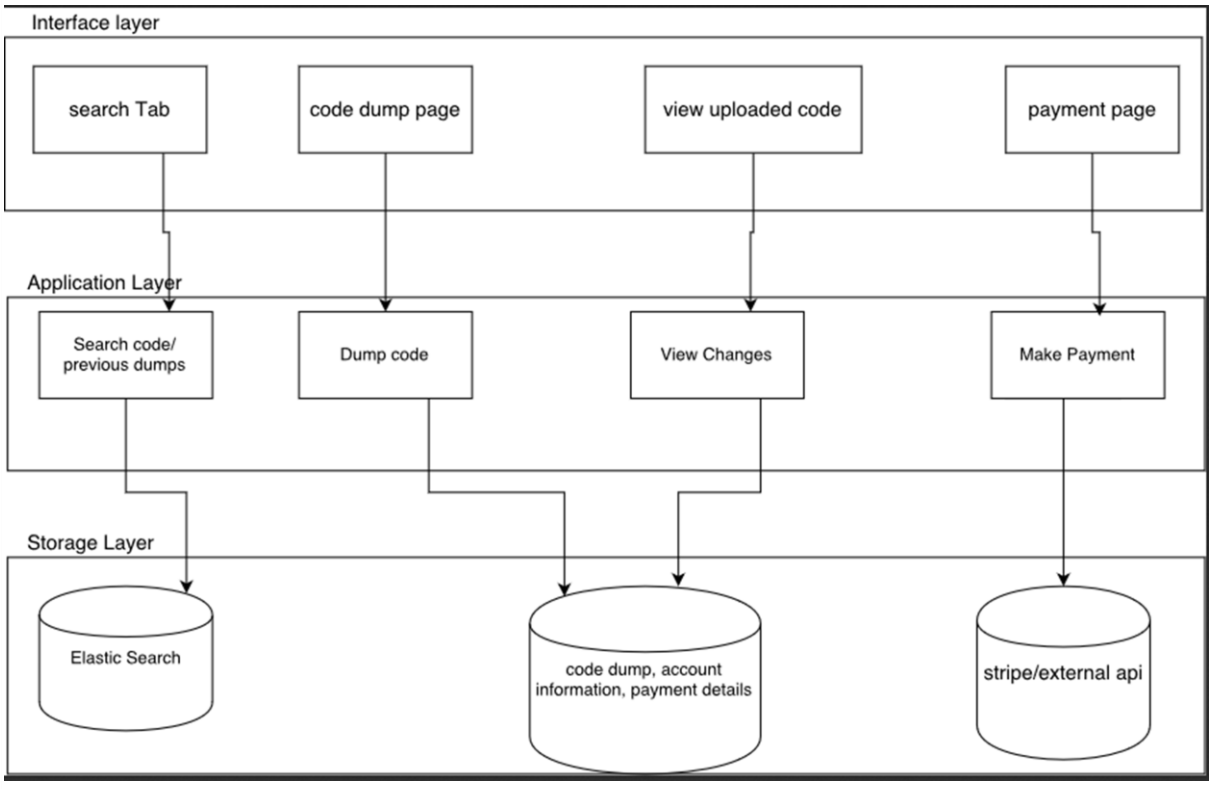
For the website to feel efficient and not lagging or slow for our market these non-functional requirements are our targets

- HTML5 and CSS3 compatibility
- Page load time within 400ms (excluding connection limitations)
- Sign up on the website within 3 minutes
- Secure PCI DSS compliant payment methods
- Code upload of any size (free within reasonable limits - 100MB or more for a fee)
- Email notification when bounty was accepted or completed
- 95+% Service availability

System Architecture

Customer user

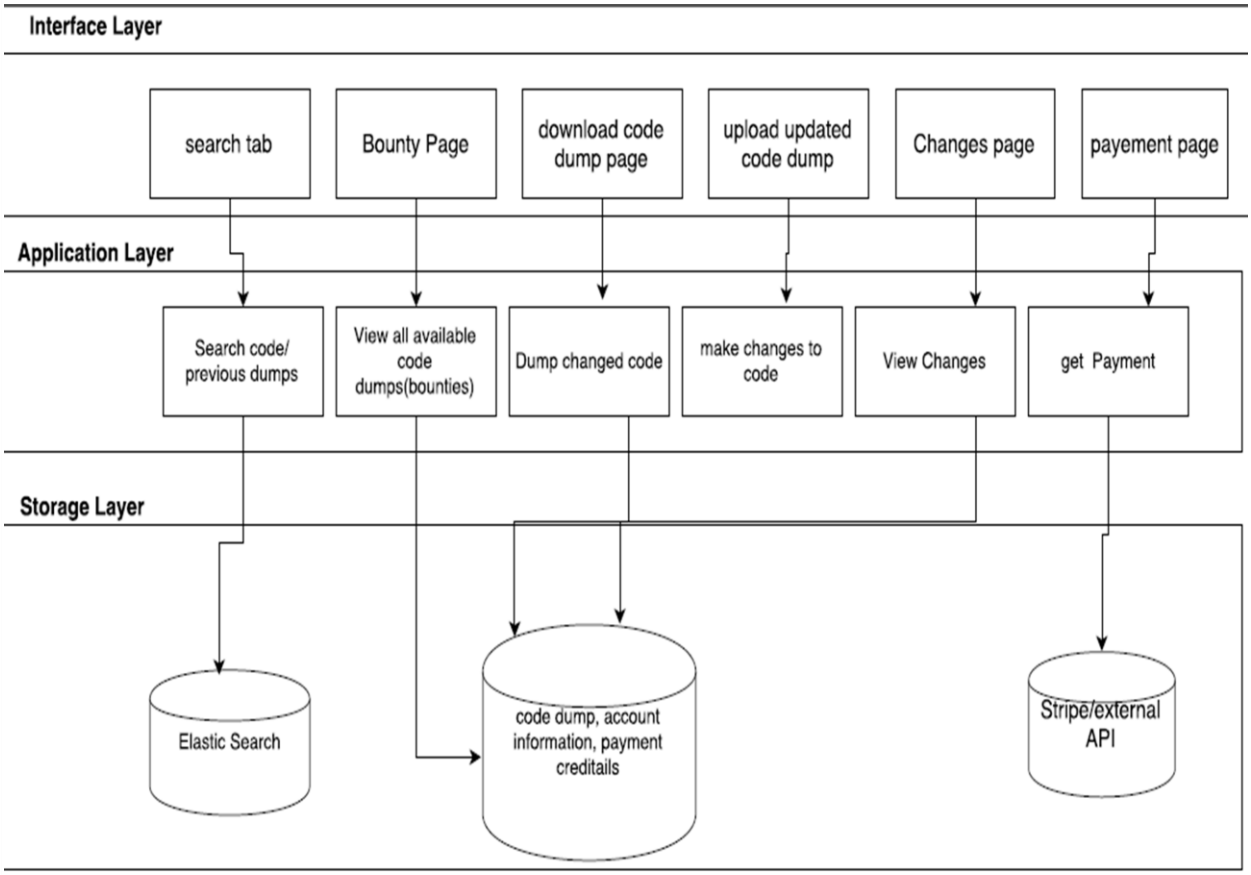
This box line diagram shows the 3 tier architecture of the customer side of the users.



This user will have less admin rights as the only need to be able to

- Log in
- Dump code
- Offer bounty on code dump
- Receive notification of status of their bounty
- View changes
- Download debugged code
- Pay the developer user

Developer user



- Log in
- View all available bounties
- Accept bounties
- Download code dump
- Make changes
- Re-upload code dump
- Pass checks
- Wait for payment
- Receive notification of payment
- Check payment
- Notification to new bounties being added to the side

Limitations

It is fair to say the technology we have chosen and the database managements system both have their own disadvantages and limitations that will in turn affect our system.

Limits Ruby on Rails

- When compare with other web based projects ruby has a slower runtime compared to nide.js or GoLang. This is not likely to create a bottleneck as we run our system on
- Depending on the number of gems dependencies and files the boot speed of rails is affected this can be a significant amount time
- It may prove difficult to fight cheap hosting (can use heroku and openshift)
- If the website becomes popular ruby does not scale well! But this is solved with a solid architecture

Limitations of PostgreSQL

- Usability and learnability of the can be difficult for beginners. But the use of heroku postgres.app or heroku postgres as a learning aid to avoid install and config problems.
- It is considerably slower than other management systems (MySQL)
- Does not support the entire ANSI SQL 92' standard,
- Less mature replication software

Prototype to Product

Due to the nature of the way that DuckDebug will function, in order for the product to properly transition from being a prototype to a fully functional system, it will require an active and diverse user-base. Evidently that will not, however, be the case upon release. In order to move successfully from a barebones system to something truly useful to you, initially there will be a small team implemented in order to ensure the system works as required. This will involve using the system as intended to ensure all its features work correctly. Stress tests and bug finding will also take place in this phase to ensure DuckDebug is rolled out as smoothly as possible.

After this, the system will finally be released to the public. Advertisements to universities and companies - both hotbeds of computer engineers, will ensure there is an initial supply and demand in terms of jobs, so regardless of whether you're looking to earn some extra income or wish to save time and money by having a bug ironed out of your code, DuckDebug will have something to offer.

After a period of time has passed, the user base will grow exponentially, and this is when the system will really start to shine. A larger user base allows DuckDebug to best match jobs to prospective users. This allows for the most complex and important jobs to be solved as quickly as possible by the most experienced engineers. It allows more inexperienced programmers that still have knowledge to offer, to solve problems at a level applicable to them. It will also allow for a more tailored experience as users begin to solve more jobs and the system adapts to their preferences.

Accessibility

Programming and programming systems have a certain disposition to being very expert-centric. This means that usually takes quite some time before the average user is comfortable with the system, understands all of its features, and is able to get the most out of their time using it. We are also aware that sometimes people are very technically skilled when it comes to creating programs and solutions in their selected field; take some business based programming languages for example - but not so much computer literate when it comes to navigating websites and using systems that they're not wholly familiar with.

DuckDebug aims to be as accommodating as possible and takes a friendly approach in order to minimize all these issues. The interface for our system is straightforward and easy to understand and approach even if it is your first time using it. We label all links based upon their content and try to avoid ambiguity, so if a button says 'Your Profile', it will take you to your profile page and there is no question on what it might do.

We are also aware that as DuckDebug is open to the public sphere, it's inevitable that it will be used by people with varying disabilities and impairments. Our system therefore has been designed so that it is easily navigated by the vision impaired with a large enough interface. Care has also been taken to integrate varying different colour schemes that will help cater to those who are colour blind to varying degrees. This is also a feature which will have further use to everyone as it has the possibility to increase readability of the entire product.

One of the main incentives we have for accessibility is time. The main reason we envision a large proportion of people using our service is to save valuable time but also money. In a business situation, every second counts and if a user has to spend half an hour working out how to successfully use and upload a job to our system, that's thirty minutes of wasted time and cash and possibly unhappy customers. DuckDebug makes it easy to upload a portion of your code in just a few clicks and get it out to as many experienced people willing to help as possible, saving you money and stress in the process.

Adaptation

How will DuckDebug fit into the future market with advancing technologies and be able to keep up with the rapidly changing face of computing?

Happily, due to the methodology we will be using when it comes to submitting and completing jobs, DuckDebug should evolve comfortably with new technologies. A large majority of the code will be submitted as files in their designated languages. The main problem arises with the search and sorting algorithms when it comes to newly developed and adopted languages being used. When submitting code, there will be an option for other languages if it is not written in a currently supported one. When a new language takes off and becomes mainstream, it will be handled accordingly using roughly the methodology below:

- Many users request a language be added or it gains popularity quickly
- Supporting the language is petitioned to the public
- Newly added language will be tested internally for a short period to ensure it works correctly with algorithms
- New version of system is released to the public allowing them to choose the language as a specialisation and submit jobs in the specific language

- Data and feedback is gathered to ensure the newly updated system works well in large scale
- Any necessary patches are developed to ensure system now runs smoothly
- Continued monitoring of how the update works with the newly added language until such a time as it is considered to be working adequately by users

Future Development

As DuckDebug grows, it is likely to become a popular method of quickly solving both minor and major bugs. A lot of the growth will be subject to word of mouth which is likely to cause exponential growth, resulting in a better user experience as jobs will be completed faster and in a more expert manner. Our sorting algorithm will lend more of the correct people to being assigned the correct jobs which will increase viability of the service being used more and more in big business. This revenue will, in turn, attract more prospective experts in each field as there becomes a bigger potential for personal profit in the business.

The more of these big companies that adopt our service; more revenue in turn can be spent on improving DuckDebug as a whole. It will allow us as a company to expand our infrastructure, reducing strain on the system and also continue development on the sorting and searching algorithms to increase quality of life for you, our users. This extra income can also be spent on invaluable support for the system as another bonus. Dedicated staff would be able to offer fast assistance to any issues taking place for users and will be able to guide confused clientele through usage of the system should they need it.

One of the large qualms users tend to have with online services is possible downtime that can be caused by attacks or glitches. Having a paid team on hand in order to keep track of the website and also complaints will allow us to catch these issues as early as possible and maintain maximum possible uptime, minimizing inconvenience.

Work Cited

- Rory MacDonald Founder and Director Founder of Made. Background in Design, Technology and Software Engineering. @rorymacdonald. "Made." Pros and Cons of Ruby on Rails - Made Tech Blog. N.p., 2015. Web. 24 Nov. 2016.
- "What Are the Pros and Cons of Ruby on Rails?" What Are the Pros and Cons of Ruby on Rails? - Quora. N.p., n.d. Web. 24 Nov. 2016.
- "What Are Pros and Cons of PostgreSQL and MySQL?" What Are Pros and Cons of PostgreSQL and MySQL? - Quora. N.p., n.d. Web. 24 Nov. 2016.
- "Mysql Vs Postgresql Comparison - Web And Dedicated Hosting Tutorials By Anchor". Anchor.Com.Au, 2016, http://www.anchor.com.au/hosting/dedicated/mysql_vs_postgres#head-0ca91f0bef42138084fa05a13d8ec1e4107d6dbd-2