

Lost&Found - University of Bristol!

1st Jinyang Zhan
Department of Computer Science
University of Bristol
Bristol, United Kingdom
qi18235@bristol.ac.uk

2nd Xiangtian Zheng
Department of Computer Science
University of Bristol
Bristol, United Kingdom
email address

Abstract—We built a small website for Lost and Found, students can post information or collect information on this website, which makes it easier to find what they lost and return what they found.

The application can be run online at <http://lostfound-env.bqbw3wcub.us-west-1.elasticbeanstalk.com>
[git@github.com:JinyangZhan/COMSM0010.git](https://github.com/JinyangZhan/COMSM0010.git)

Index Terms—Lost and Found, Website, AWS

I. INTRODUCTION

In university, students have lots of precious things such as ID card, phones and credit card. If we lost these things, it is hard to live a normal life in this highly informative society. At the mean time, its hard for us to find things we lost by ourselves, even if we can achieve this, it is completely inefficient, for the reason that the information we can get is limited.

For example, a student Bill left his Ucard at the study room of Queens Building but he didn't realize it until he arrived home. Unfortunately he can't remember where he lost the property and got no idea. He would waste another 20 pounds and more time on it if he can't find it.

But things will go different if we have a platform to post what we lost and ask for help. Bill could post what he lost online and some kind friends would help him find it. On the other hand, those people who find something lost do not have to worry about how to find the owners. They can simply post what they find on the website, once the owners log in our website, they will see it.

This website is built for these kinds of situations. We offered this platform for students in University of Bristol, which is easy to handle. This website realizes centralized loss of property claiming.

The website is inspired by my own experience and for our students who enjoy the development of internet, a website is obviously more efficient than a "Lost and Found" office.

II. MOTIVATION FOR THE USAGE OF CLOUD COMPUTING

Cloud Computing is the dynamic allocation of on-demand computing resources to external customers over the network, which can be compared to other utilities like electricity or gas. You do not care about where you can get the computing resources or how to manage it, you just compute how many resources you need and then apply for it. As a generalised

term, Cloud Computing usually relates to large-scale distributed computing, parallel computing and network storage technologies, etc.

One benefit of public cloud computing is that people can use massive amount of compute power at any time without actually paying for the operation and maintenance costs of these resources, which is powerful for individuals or small companies who do not have the budget or need to buy their own hardware.

Also, Users can connect to the extended service over the Internet to meet their needs, customize applications and access cloud services from anywhere. Enterprise users can quickly bring applications to market without worrying about the underlying infrastructure costs or maintenance. Also, cloud services provide enterprises with a competitive advantage by providing the most innovative technologies.

III. IAAS, PAAS & SAAS

A. IaaS

IaaS (Infrastructure as a Service) is the first layer of Cloud Computing, sometimes also called Hardware as a Service. If you want to run some enterprise applications on the office or company website a few years ago, you need to buy servers or other expensive hardware to control the local applications to run your business.

But now with IaaS, you can outsource hardware to other places. IaaS will provide off-site servers, storage and networking hardware that you can rent. Save on maintenance costs and office space, companies can use these hardware to run their applications at any time.

B. PaaS

PaaS (Platform as a Service) is the so-called middleware. All of your company's development can be done at this level, saving time and resources.

PaaS offers a variety of solutions for developing and distributing applications, such as virtual servers and operating systems. This saves you money on hardware and makes it easier to work with dispersed studios. Web application management, application design, application web hosting, storage, security, and application development collaboration tools.

C. SaaS

SaaS (Software as a Service) is the layer that comes into contact with your life every day as a web-based services. Any application on a remote server can run over the network, which is SaaS.

The services you consume are entirely from web pages such as Netflix, MOG, Google Apps, Box.NET, Dropbox or Apple's iCloud. Although these web services are used for business and entertainment or both, they are part of cloud technology.

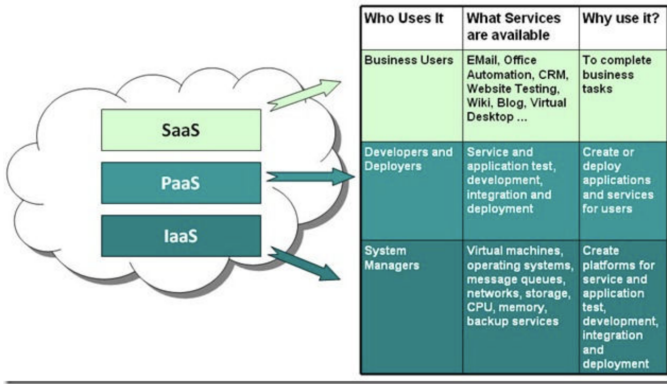


Fig. 1. Definitions from NIST

D. Our Choice

As a lightweight small website, we do not need massive computing resources or some specific system requirements, so we choose PaaS to host it. We chose AWS Elastic Beanstalk to host the website because AWS provides highest memory size and CPU clocks, also it provides initial credits which valid for 12 months.

1) *AWS Elastic Beanstalk*: Elastic Beanstalk is Amazon's PaaS service for deploying and scaling web applications and services. With Elastic Beanstalk, you can upload your code and automatically handle the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. Our Lost&Found uses Elastic Beanstalk to deploy the web application and then it would automatically build the selected supported platform versions and provision one or more AWS resources to run the application [1].

IV. WEB APPLICATION

Lost&Found is written in PHP, which is a powerful server-side scripting language for creating dynamic interactive sites [2]. PHP can do a lot more than just develop a server-side script. One of the most powerful and significant features of PHP is that it supports a wide range of databases. It is very simple to write a database-backed web page using any extension to a database (such as MySQL), or use an abstraction layer such as PDO, or connect to any database that supports the ODBC standard through an ODBC extension. Lost&Found uses PDO library to connect a MySQL database.

A. Demand Analysis

With the increasing complexity of software systems and the expansion of scale, demand analysis has become more prominent in software development. The so-called demand refers to the users requirements on the target software system in terms of function, behavior, performance design constraints. In the research of demand analysis, system developers must investigate and analyze the needs of users, and correctly define the characteristics of target software based on the characteristics of the actual environment. Therefore, demand analysis is critical.

1) *Administrator*: As administrator of the website, admin user can manipulate posts and comments, modifying or deleting them. Admin user also has privilege to manage regular users (as Fig 2).

2) *Regular User*: Students can register as users. Users can post and comment to find what they lost or return what they found.

3) *Visitor*: If students don't want to register, they can visit the website as visitor. The only thing that a visitor can do is to look up posts.

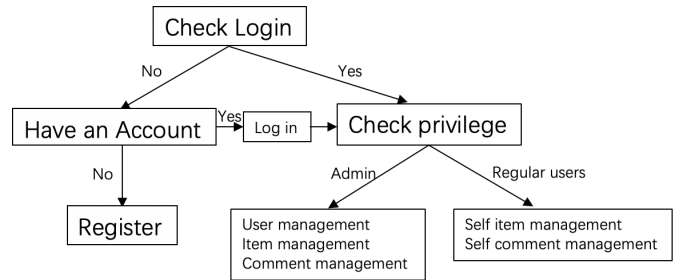


Fig. 2. A simple process of user account

B. Database Design

Lost&Found deploys MySQL Database. There are three tables to store data of items, posts and users in the database. One thing that should be noticed is that all the deletion operations are pseudo-delete, i.e. set the parameter disabled equals to 1 (as Fig 3).

C. Implementation

1) *User Account*: When a visitor first visit Lost&Found, he has no privilege to post or comment. He needs to register as a new user. Admin user is specified as a user who can manage regular users, items and comments. Resiter.php is used to process registry, using Database::conn() to build connections to the database and inserting a new user object in the MySQL table when a series of parameters are passed (as Fig 4). Password needs to be salted encrypted to ensure the reliability of password storage.

Fig 5 shows how back-end database controller process login requirement. When visitor clicks the button "Log in", he will jump to the login page and be asked to fill in username and password. This would pass in a POST request with username

Field	Type	
id	INT	↕
lost_or_found	TINYINT	↕
item_name	VARCHAR	↕
description	TEXT	↕
publisher_id	INT	↕
publish_time	TIMESTAMP	↕
view_count	INT	↕
reply_count	INT	↕
closed	INT	↕
disabled	TINYINT	↕

(a) User table for users

Field	Type	
id	INT	↕
lost_or_found	TINYINT	↕
item_name	VARCHAR	↕
description	TEXT	↕
publisher_id	INT	↕
publish_time	TIMESTAMP	↕
view_count	INT	↕
reply_count	INT	↕
closed	INT	↕
disabled	TINYINT	↕

(b) Item table for items

Field	Type	
d	INT	↕
tem_id	INT	↕
poster_id	INT	↕
post_time	TIMESTAMP	↕
content	TEXT	↕
disabled	TINYINT	↕

(c) Post table for comments

Fig. 3. Three tables in database to store data

[Back to index](#)

Registration

Username:

Nickname:

Password:



Register

Return to login!

Fig. 4. Register an account on the website

and password. Then a link to the database will be connected, searching the database for a user who matches such a username and password. If there exists such information, the login is successful, jumping back to the homepage, otherwise an error is displayed.

```

if (checkLogin()) {
    header("location: index.php");
}
else if (isset($_POST["username"]) && isset($_POST["password"])) {
    // Connect with database using own class
    $auth = Database::conn();
    // SELECT if there is a user which matches username and password
    $login = $auth->search(
        "SELECT * FROM user WHERE username=? AND password=? AND disabled=0",
        array($_POST["username"], encrypt($_POST["password"]))
    );
    if ($login) {
        $_SESSION["user_id"] = $login["id"];
        $_SESSION["user_name"] = $login["name"];
        header("location: index.php");
    }
    $flag = true;
}

```

Fig. 5. The implementation of login check

2) Fundamental operation:

- The Item Modify page must be logged in to access. If

the parameter contains rid, which is the parameter of the deleted item, the permission will be detected, and if the permission is satisfied, the item is deleted. If the parameter contains itemname, description, and lof, it may be a new item or a modified item, and then check whether the status and id parameters are included. If yes, it is modified, if not, it is added. If you include an id, it must display the user interface.

- In order to distinguish between admin users and regular users, we need to check privilege when a user is going to modify or delete items or comments. Administrator would have all the privileges to modify posts, comments and users. If not, check whether the user is the publisher of the post, if so, the user can operate the post, otherwise check whether the user is the publisher of the comment, if so, he can operate the comment (as Fig 6).

```

// Check if is privileged
function checkPrivilege($item_id = NULL, $comment_id = NULL) {
    $user_id = $_SESSION["user_id"];
    $conn = Database::conn();

    // Manager has full privilege
    $user = $conn->search("SELECT manager FROM user WHERE id=?", array($user_id));
    if ($user["manager"]) {
        return true;
    }

    // If user is the item publisher?
    if (isset($item_id)) {
        $item = $conn->search("SELECT publisher_id FROM item WHERE id=?", array($item_id));
        if ($item["publisher_id"] == $user_id) {
            return true;
        }
    }

    // If user is the post publisher?
    if (isset($comment_id)) {
        $post = $conn->search("SELECT poster_id FROM post WHERE id=?", array($comment_id));
        if ($post["poster_id"] == $user_id) {
            return true;
        }
    }

    return false;
}

```

Fig. 6. Check privilege when user tries to operate posts or comments

V. DEPLOYMENT

A. Web Architecture

The infrastructure that Lost&Found used is listed as below:

- 1) AWS EC2: auto-Scaling features used for web servers and GPU work servers.
- 2) Tencent RDS: MySQL database for storing user login information, and details about items and comments.

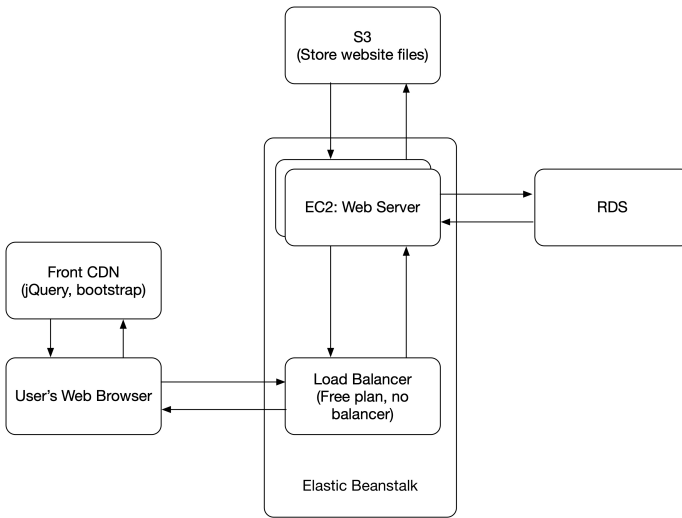


Fig. 7. A brief architecture of Lost&Found

- 3) AWS S3: Data generated by users and content of the website are stored in S3.
- 4) AWS CloudWatch: used by Auto Scaling Group to trigger scaling operations.

A brief architecture of Lost&Found which we implemented is shown as Fig 7.

B. Deployment Implementation

1) *Create and connect to an instance:* First, to deploy the Lost&Found website to the Elastic Beanstalk, we need to create an environment. We logged in to the AWS Management Console and choose "Build a web app with Elastic Beanstalk", then choose PHP as the platform and upload the code from local computer. After an environment being created, the Elastic Beanstalk would automatically create an EC2 instance, an Elastic Load Balancer.

2) *Configuration:* Change the environment type from Single Instance to Load Balanced to set Auto Scaling to optimize the number of instances used. Also it can specify which AZs can be used.

3) *RDS:* Instead of using AWS RDS, we chose Tencent RDS as we used it previously. AWS RDS is also simple to implement, you just need to modify Database, changing Engine to whatever database you need. As a lightweight website which is considered not have too much dataflow, we just use MySQL to store data.

VI. SCALABILITY

The scalability of Lost&Found is mostly implemented automatically by AWS Beanstalk. Unlike operating in physical environments where if you need to scale up or scale out, you need to either invest in powerful hardware to meet the demands of increasing business or follow a distributed model of investment. By moving the website application to the cloud, we can take advantage of the elasticity of the cloud to tightly integrate the use of infrastructure with our needs.

Elasticity optimizes resource acquisition and release, allowing your infrastructure to rapidly scale in and scale out as demand fluctuates. To implement elasticity and scalability, configure the Auto Scaling settings to scale up or down based on metrics from the resources in your environment (utilization of the servers or network I/O, for instance). Also, it would be powerful to publish system metrics (CPU, Memory, Disk I/O, Network I/O) to Amazon CloudWatch and configure alerts for triggering Auto Scaling actions or sending notifications. [3]

VII. FUTURE WORK

- **More functions:** We can add social features to the website. Users can send messages to other users when they need further information about their posts or comments. We can further allow users to insert images to their posts, helping people quickly figure out whether the item is what they lost.
- **Security:** We should further consider about security of the website, finding out shortcomings and fixing it.
- **Change the design of some workloads:** The first step is to introduce S3 and CloudFront. Move static content from the Web instance to S3. Suitable file types include static data (CSS, JS, image, video), log, backup, etc. S3 is massive storage and can support a large number of concurrent access, the cost is also very low. On the CDN side, CloudFront and Web service interface provide services, support dynamic and static content, streaming video, support root domain, and support customized SSL certificates.

The second step can be to introduce ElastiCache and DynamoDB. ElastiCache is the hosted Memcached and Redis service with the same API, both are very fast cache services (millisecond level). The difference is that Memcached uses one AZ, Redis can copy across AZs. DynamoDB is a NoSQL service with background storage based on SSD and an average latency of milliseconds.

VIII. CONCLUSION

In this paper, we have presented Lost&Found - a website for students of UoB to find lost items. The web application is hosted on AWS Elastic Beanstalk, utilising several different AWS technologies in order to make it scalable.

Although we have implemented fundamental functions of the website, it has much more work to be developed in the future to improve the whole users experience.

REFERENCES

- [1] What Is AWS Elastic Beanstalk? - AWS Elastic Beanstalk. (2019). Retrieved 18 August 2019, from <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html>
- [2] PHP: Hypertext Preprocessor. (2019). Retrieved 18 August 2019, from <https://www.php.net/>
- [3] Bellenger, D., Bertram, J., Budina, A., Koschel, A., Pfnder, B., Serowy, C., ... & Schaaf, M. (2011). Scaling in cloud environments. Recent Researches in Computer Science, 33, 145-150.