

CSCI 3172 Web-Centric Computing

Project Proposal

Guest Wi-Fi Management System

Yanlin Zhu B00812966

Xinhai Cheng B00796386

Team Member

Yanlin Zhu, B00812966, yn842496@dal.ca

Skills: Java, C, Python, HTML, CSS, JavaScript, PHP, MySQL, Android Studio, Vue, React, Node.js, Laravel, Git, Agile development

Experience:

1. Participated in the team project of Vue+Laravel+Mysql to design a student survey management website for Dalhousie University School of Health and Human Performance. The website can automatically generate downloadable reports by uploading, storing and analyzing student surveys. Participated in database design, mainly responsible for front-end interface development, login registration and front-end authentication control.
2. Developed bill tracking app based on Jetpack MVVM architecture with Kotlin to implement bill history tracking, automatic chart and report generation function

Xinhai Cheng, B00796386, cqh@dal.ca

Skills: Linux, Network, CISCO, MySQL, Oracle, Ansible, Openstack, VMware, Java, PHP, Laravel, HTML, CSS, JavaScript, Git

Experience:

1. Participated in the construction project of PaaS cloud platform of China Telecom, completed the interaction of hardware and system configuration. Through using Ansible as the batch management platform, I completed the cluster deployment of hundreds of servers. Based on Redhat Enterprise Linux

and OpenStack, this platform provides convenient, real-time delivery of public cloud computing services to millions of people and enterprise users.

2. Participated in the development of cloud-based wireless network hardware products, and realized cloud takeover of all devices through TR069 protocol, configuration delivery and health check. This hardware product can help small and medium enterprises quickly develop and manage the network anytime and anywhere. It has now helped dozens of Small and medium-sized Halifax local businesses to build faster and more stable wireless networks.

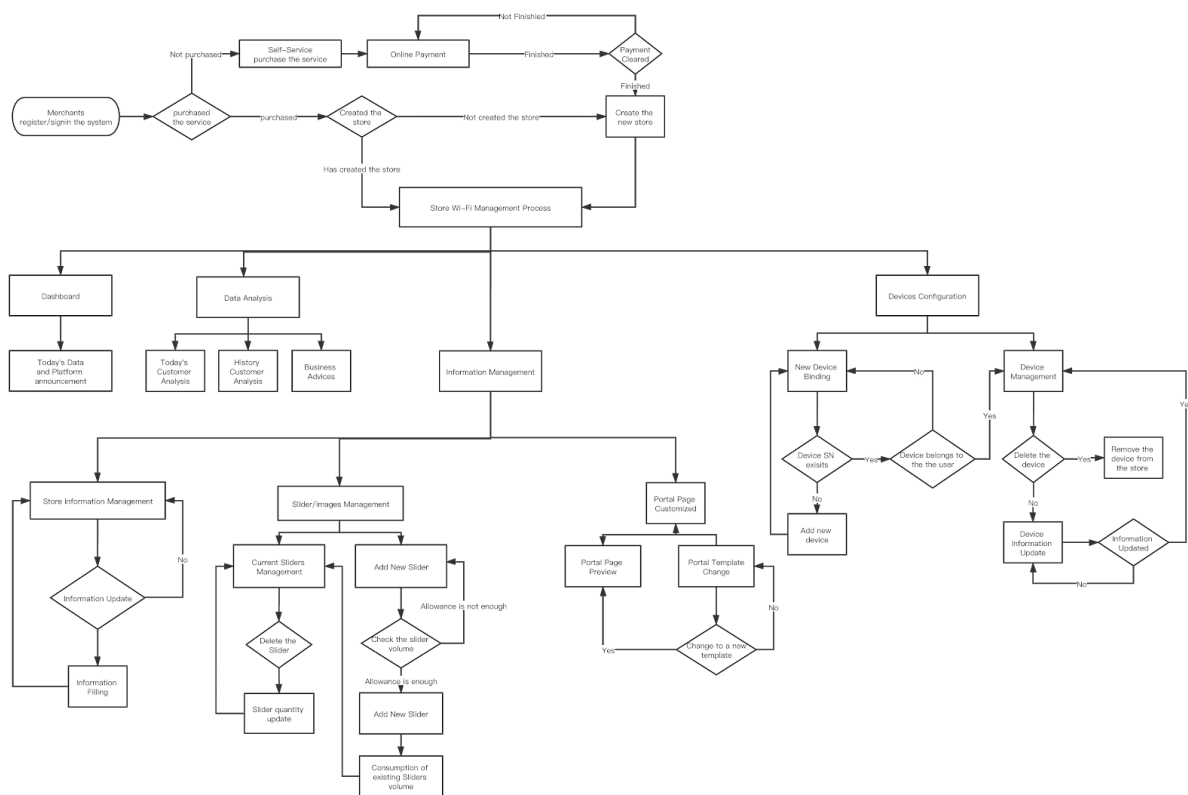
Summary of Application

Guest Wi-Fi Management System is a Platform as a Service, which combines with multi-vendor hardware to provide customers with public service network service capability. This system can both provide vertical customers (merchants) with cloud-based online network management and information publishing capabilities, and horizontal customers (merchants' customers) with wireless network authentication, Internet access services and information services.

Based on Guest Wi-Fi Management System, merchants can manage their store network hardware and authentication page anytime and anywhere to help them better understand and control the network. At the same time, Guest Wi-Fi Management System can also help

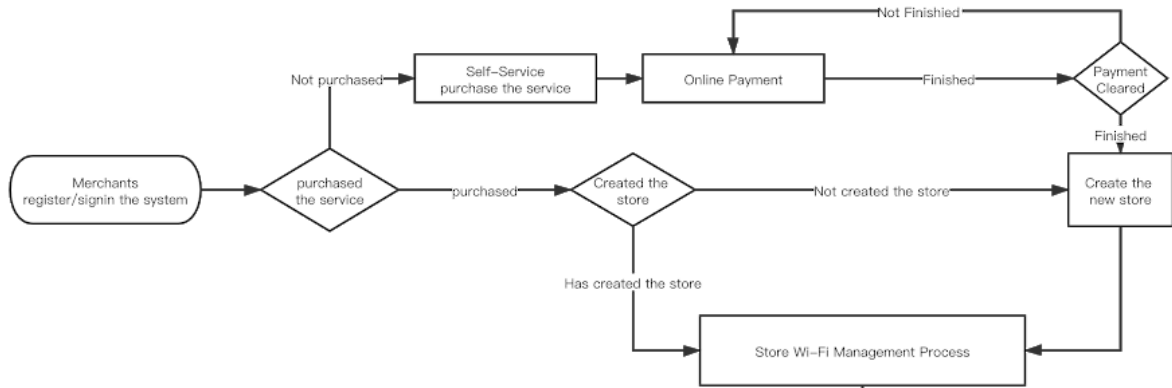
merchants complete business optimization and data analysis services by properly collecting user authentication information. Based on the Guest Wi-Fi Management System, merchants can also enhance their brand building, release important information and promote their business by customizing the wireless network authentication page. All these functions can help merchants maximize the utilization of wireless network resources and implement the business value based on wireless networks.

Light Use Case:

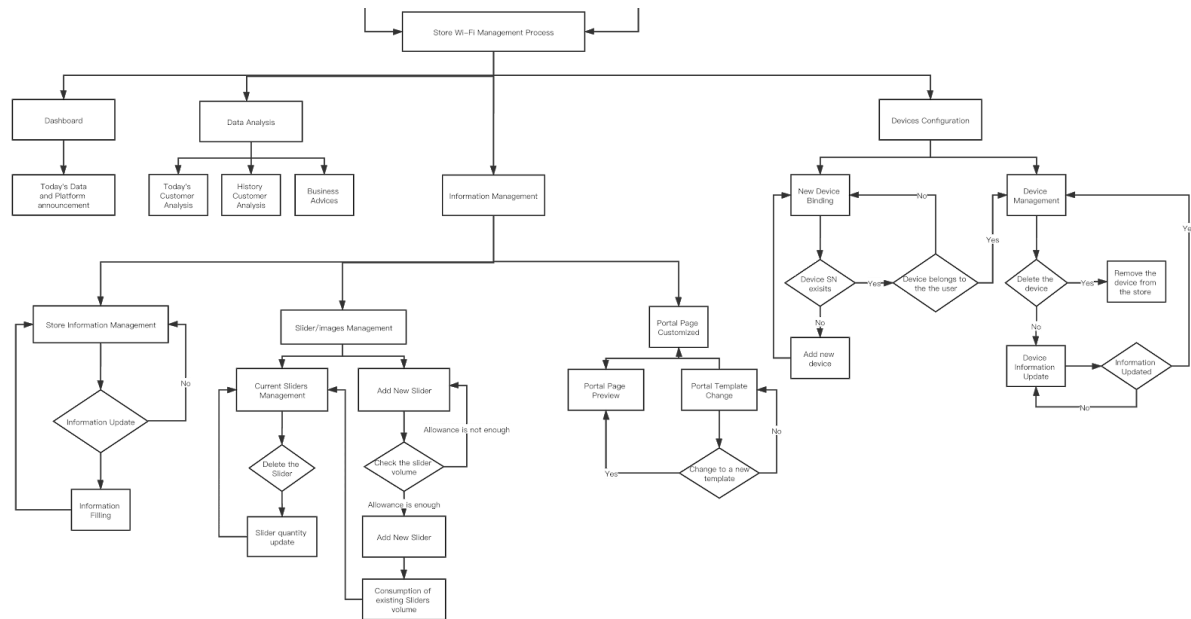


Dashboard Operation Process

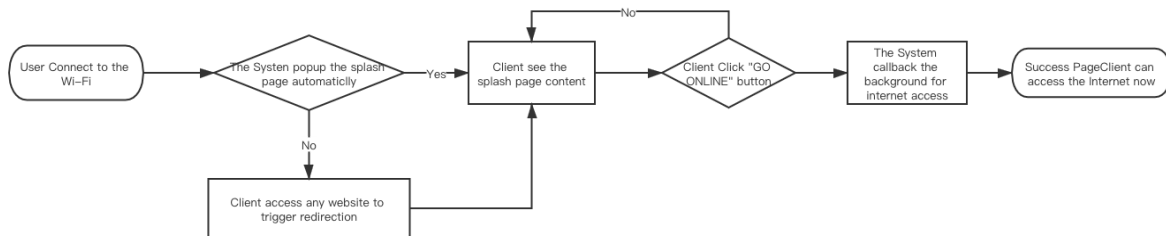
For a Vertical Customers (Merchants):



As shown in the figure, the first part is the user account and service packages check module. When the merchants complete the registration in the system, they can sign in to the merchant dashboard. The system will inquire the database to know whether the current user has the service purchase history or the store information that has been bound in the account. If this is the first time a merchant has registered or has not ordered any services, the system will automatically redirect the user to the service purchase page. The user selects the service plan they need and completes the payment through online payment. The system obtains the payment status of users through third-party payment platforms, such as Stripe or Paypal, and allocates corresponding packages for users. After the user opens the service package, the system will provide binding guidance for customers who are not bound to the store information, and help users fill in the store information and access the device by using Bring your own device (BYOD) method.



After completing the configuration of store information or equipment, users can use all the functions of the system. Users can view the latest announcements through the merchant dashboard to ensure that they can know the information and system status. At the same time, users can use the visitor data analysis function to view the big-data of current Wi-Fi visitors in real time. In addition, users can view the recent historical data and view the business optimization information given by the system. The information management function can help users to view and update the store information in real time. At the same time, users can also delete and upload pictures on Wi-Fi authentication pages to help merchants better promote their products. All information changes can be previewed through Portal page management, and templates can be switched through the Portal page management function.

For a Horizontal Customers (Merchants' Customers):

When a horizontal customer (the merchant's customer) arrives at the merchant's store to consume and use the network, it will first turn on the device's Wi-Fi Settings, select and connect to the merchant's public wireless network. After connecting to the network, with hotspot protocol, the system will automatically pop up the splash page window. If the system does not have a pop-up window or the user accidentally closes it, the user needs to reconnect to Wi-Fi or open the browser to access any website to be redirected to the splash page. Once the users are redirected to the splash page, they can complete the Internet connection operation by clicking the "GO ONLINE" button on the page. After the user clicks the button, the system sends the authentication request to the back-end service, and the back-end service receives the request and forwards it to the device. Then the user will be redirected to the success page, so that can browse the Internet normally.

Security Risk:

1. Information Data Storage Security

The security of user information and data is very important, which relates to the privacy of users and the commercial interests of merchants. For the storage of user information, we will choose encrypted storage and permission management methods at the system level. The user password will be stored with MD5 and Salt to ensure that the user's password is secure even in the worst case of database leak. At the same time, we will encrypt the whole website during deployment and protect users' data from illegal access through Transport Layer Security (SSL).

2. Online Payment Security

Users' payment security is also a challenge due to the online ordering capability of the site. In order to improve the payment security level of users, we do not store any payment information of users locally, such as credit card number, cardholder name, CVV, etc., so as to ensure that even in the worst case, the database is leaked and the attacker cannot obtain the payment information of users. At the same time, we use third-party payment companies like Stripe and Paypal for payment settlement. In the case of increasing user trust, we can also improve the overall security of the payment system through RESTful API provided by third-party companies.

3. Information Exchange Security

Due to technical limitations, the use of authentication in Portal pages requires the collection of MAC address, IP address and other information of users, as well as the

security of information exchange when the device collects and processes user data. At the same time, network hardware devices also need to contact and authenticate with the Radius server, so that users can complete network access operations. In the process of network hardware and server connection, information exchange security also needs to be paid attention to. We will process the page URL by encrypting the user's MAC and IP address. When the user's device takes the initiative to send such information, the system will generate random Token through the encryption algorithm to replace the user's real MAC and IP address. Even if the user's TOKEN leaks, it will not be affected. At the same time, network devices will be connected by RADIUS over TLS (RADSEC) to ensure that user information will not be intercepted in the process of transmission to the server.

Tech Stack:

We decided to use Vue.js, Element UI, Laravel and MySQL for our project.

Frontend Framework



| | | |
|--|--|--|
| 2010 by Google | 2014 by Evan You | 2013 by Facebook |
| Google, Wix, Weather.com | Gitlab, Alibaba, 9gag | Whatsapp, BBC, Twitter |
| real DOM - slow | Virtual DOM - fast | Virtual DOM - fast |
| Backed by Google | Backed by open-source community | Backed by Facebook |
| Size: ~500KB | Size: ~80KB | Size: ~100KB |
| Heavy-weight apps | Light-weight apps | Light-weight apps |
| Routing: Inbuilt | Routing: External libraries(React Router) | Routing: External libraries (vue-router) |
| Template and typescript logic are separate | Template and typescript logic are separate | Everything in one jsx file |

We first exclude Angular from our consideration since it is a heavy weight framework compared to the rest two. Also, none of our group members are familiar with Angular. When it comes to React and Vue, the reason why we choose Vue over React is its simplicity. Vue retains the separate writing method of HTML, CSS and JS, so we can keep our original habits when developing. The template is ordinary HTML, the data binding uses Mustache style, and the style uses CSS directly. While React does not have a template, it is just a render function that returns a virtual DOM tree. The way that React recommends is JSX + Inline Style, which is to write both HTML and CSS into JavaScript. Vue is more suitable for us in terms of learning cost. Also, Vue is helpful for its two-way data binding and scoped CSS.

As we choose Vue as our frontend framework, we will use a set of libraries to help build the project.

Vue CLI: A quick scaffold to build this single page application.

Vue-router: A Vue's routing manager that manages component routing

Axios: The promise-based HTTP library responsible for data transfer between server and client side.

Vuex (optional): A centralized management solution for Vue to centrally store and manage the state of all components of an application.

UI library



| | |
|---|---|
| 2016 by ANT Group (Split from Alibaba Group) | 2016 by ELEME (subsidiary of Alibaba Group) |
| Initially designed for React, support Vue since 2018 | Focused on Vue |
| Focus on enterprise-level management platform development | Designed for general development |
| more complex | easy to learn |

We decided to use Element UI since it is focused on Vue while Ant Design is focused on React and its Vue version is just a simplicity version. However, this is not a final decision, we

may use Element UI for general design and Ant Design for backstage management platform design.

Backend Framework



| Django Software Foundation | Taylor Otwell | Node.js Foundation |
|----------------------------|--------------------|-----------------------------|
| 2005 | 2011 | 2009 |
| Python | PHP | Javascript |
| MVT(Template) | MVC | / |
| Rapid Development | Clean Architecture | Scalability and Performance |

As for the backend framework, there are three options available: Django, Node.js and Laravel. Django is built by Python which is easy to learn for Python users. However, we only have a basic level of Python experience, therefore, it is excluded due to learning cost. Also, we exclude Node.js for the same reason as Django and React – we are not experienced with JavaScript. The reason why we choose Laravel is we are familiar with PHP. As all our group members are third-year students who just finished CSCI 2170 – Server-side Scripting last term, we are familiar with PHP development. As none of us is familiar with other backed frameworks, Laravel becomes our best and only choice. Another point that attracted us is its MVC design pattern which is also a most familiar design pattern for us. It is easy to use and thus offers an extremely convenient way to build projects. The Laravel framework provides

an Artisan built-in command-line tool to help automate tedious and repetitive programming tasks, for example database migration.

Database

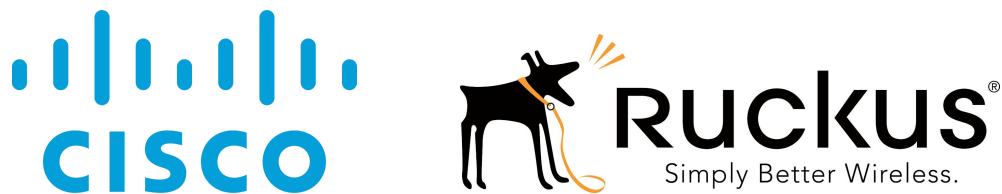


| 2009 | 1995 |
|--|--|
| MogoDB | Oracle |
| C++,C,JavaScript | C++,C |
| -Collection -Document -Field | -Table -Raw -Column |
| Dynamic Schemas | Strict Schmas |
| Document-oriented | RDBMS |
| Horizontally Scaling | Vertically Scaling |
| -High write loads -unstable schema -set to grow big -No DBA | -Data structure fits for tables and rows -Frequent updates and modifications of large volume of records -Relatively small datasets |

Finally, as for databases, we choose to use MySQL for the same reason as our backend. Since we only have two group members and the time is intensive to build such a large project,

learning cost efficiency is our priority. All members of our group are familiar with MySQL. We can use SQL queries frequently without additional training. While other competitors like MongoDB as a NoSQL database are quite difficult to learn. Especially considering Laravel is good at its ORM handling and MongoDB removed the ORM layer.

Hardware



Cisco is currently the most trusted vendor for enterprise network hardware, and after comparison, we decided to use Cisco Aironet 1800 Access Points. The product supports the latest Wi-Fi technology, 802.11AC Wave 2, which can balance the cost and demands. It can also connect with Wi-Fi management system and help users complete authentication through Portal 2.0 protocol. In addition to indoor products, we chose RUCKUS T811-CM Outdoor Access Point for the outdoor Wi-Fi coverage. The RUCKUS product is waterproof, lightning resistant, and suitable for outdoor deployment and continuous service for users. RUCKUS T811-CM Outdoor Access Point can be integrated with Wi-Fi management systems using either Portal + Radius or Portal 2.0 protocols.

Team Charter:

Weekly Meeting time:

| Time | Location |
|------|----------|
|------|----------|

| | |
|------------------------------|--------|
| Sunday: 3:00 PM - 6:00 PM | Online |
| Wednesday: 2:35 PM - 4:55 PM | Online |

Platform for discussion:

There are three major aspects of communication for our project: shared document, version control and messaging.

- Shared Document

There are two software available for us: Microsoft Word shared document and Google Doc.

These two platforms are similar and can handle most tasks properly. MS word is more powerful in terms of its functionality. For example, better table formatting and controls, more control over text styles, and better graphic and image implementation. However, in terms of collaboration, Google Doc's document sharing is simpler and more convenient than MS Word's. Google Doc documents can be accessed directly from the web, while MS Word is stored locally and can easily be confused with documents from other courses.

Therefore, we decided to choose Google Doc as our shared document platform since collaboration is the first priority of our project not paperwork.

- Version Control

There are two platform of version control available for us: Git and SVN

Git is a distributed version control which uses multiple repositories including a centralized repository and server, as well as some local repositories. SVN is a centralized version control which makes it easier for managers to have more of a top down approach to control, security, permissions, mirrors and dumps.

In this project, we will use Git since Yanlin is not familiar with SVN.

- Messaging

There are two major types of messaging platforms available for us: asynchronous communication and synchronous communication.

Asynchronous communication platforms like Slack, MS teams, are suitable for a large group to share everyone's progress especially suitable for those who work in different time zones. Also, this type of messaging will not disturb work life balance.

Synchronous communication platforms such as What's app and Wechat, are good at receiving instant reflection and reducing replying process time. It is good for a small group.

We are using Wechat as our messaging platform since we are friends and this is a small group. Using instant messages can help us reduce replying process time. Also, Wechat provides us meeting functions.

Conflict Resolve:

Once a conflict occurs, we will schedule a meeting to talk about this issue and let everyone share his understanding and then set . If we can not convince each other, one of us will make a compromise. Next time, the other one make a compromise.