

**PROJECT PROPOSAL**

(E-Wallet and QR Code Purchase Platform)

Practice Module for Certificate in Securing Ubiquitous Systems

**Team 2**

Team Members:

Xu Jiao (E0384412)

Zou Xuan (E0384799)

**PRIVATE & CONFIDENTIAL**

*Content of this document is provided as part of project stated above, and may not   
be used, reproduced or distributed except for the sole purpose of that project.*

Document History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Comments** |
| 0.1 | 21 Jul 2019 | Draft |
| 0.2 | 24 Jul 2019 | Update |
| … |  |  |
| 1.0 | 28 Jul 2019 | Final |

Document Usage

This document is confidential to the client to whom it is presented, and is not intended for public use. It must not be reproduced or transmitted in any form or by any means, in whole or in part, without the express permission of the document owner.

The functional and design specifications contained herein summarise the features of the proposed system and the deliverables from document owner to the client under the system.

1. Table of Contents

[1 Overview 2](#_Toc14615551)

[2 Current application analysis 2](#_Toc14615552)

[3 Scope of Work 3](#_Toc14615553)

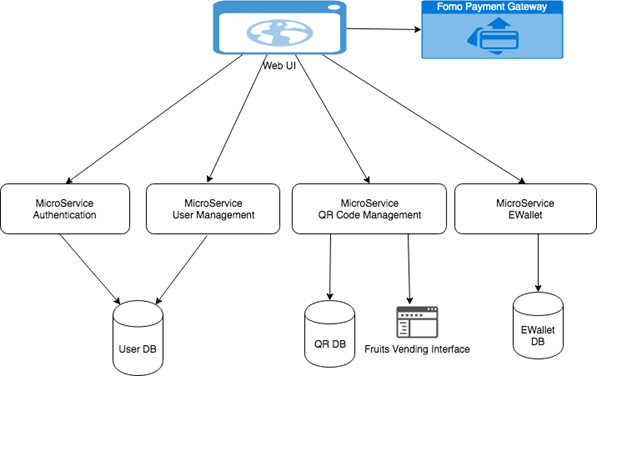
[4 Effort Estimates 4](#_Toc14615554)

# 1 Overview

A fruits vending machine company would like to establish a membership system to public customers. With this platform, public customers will be able to register as a member and top-up e-wallet, purchase QR code with e-wallet balance. The company will be able to recognize their customers and improve the quality of their service based the customer data/feedback that they will get from this platform. This requires exposing its existing internal services to the internet, the creation of suitable web-based and mobile client applications. The security architecture of the platform has to be analysed, designed and reviewed to ensure that systems, data, network and other infrastructure of the platform are adequately protected.

# 2 Current application analysis

**2.1 Current project architecture**



It is based on the last term project which have web application on microservices architecture on cloud. Currently we going to add hybrid mobile application and other security functions into this project.

**2.2 Security issue to be solved**

1) User Management

a. User Management only has simple register function, will implement 2FA to help address the vulnerabilities of a standard password only approach. With 2FA implemented, the “Forgot Password” can be added to provide better user experience.

b. User Management is not protected by any token. Even though there was no user token during registering stage, this API can still be protected by client credential token.

2) Authentication

a. Authentication service is directly connecting User DB, which makes User DB has to open a firewall to Authentication Service. This puts user data in risk if authentication service is not handling data properly. Consider provide and API in user management service to validate user name and password, Authentication service will call this API to validate user instead of directly connecting to DB.

b. Hard coded one client credential, so that “Fruits Vending Interface” and Web UI is using sharing same clientId. To solve this issue, there should be a configurable place to maintain different clientId and scope to constrain the usage of token be only applied to specific endpoint.

c. The token issued does not include user’s role info

3) E-Wallet &QR

User info is read from payload of the request instead of from the JWT token. Which can cause the risk of exposing other user’s info. To solve this issue, the E-Wallet/QR should get user info from the token, so it only returns data for that specific user.

**2.3 Other Security plan**

1) Encrypt user name, password, QR Code data during transfer and storing.

2) Obfuscate Source Code for mobile package and web application javascript

**2.4 Threat modelling**



Based on the threat modelling of hybrid mobile apps and web applications.

We going to add security controls like:

1. TLS 1.0 (network attack)
2. Data encryption (application data, user private data and authentication token/user credentials)
3. User authentication /Authorization (upgrade authentication service)
4. App store verification
5. App permissions enforcement (Will ask user for permission for every native driver usage)
6. Java script obfuscation(both mobile and web code)
7. Whitelist of sites accessible through webview.(trim access down to only those URIs that actually need to use in the app)

**2.5 Mobile platform security**

1. Stay up to date with Cordova and Ionic versions is a best practice as security fixes are included on a regular basis. Higher version provides some critical capabilities that can improve the overall security of the app. Specifically: Use Crosswalk 15 and Cordova 5+( Crosswalk is a patched and feature rich version of Chromium with access to important security features not available versions of Android as recent as 4.3 and 4.4.)
2. Use a strict Content Security Policy (disables both eval() and inline script and only allows access to JavaScript and CSS files from the same origin as the HTML page)
3. Use Microsoft Intune as a MAM/MDM tool to detect malware, and jailbreak

# 3 Scope of Work

1. Solve all the security issues addressed in part 2.
2. Add one product management page for admin
3. Use Web Component to rewrite Web UI
4. Add hybrid mobile application using ionic+cordova.

# 4 Effort Estimates

|  |  |
| --- | --- |
| Task | Efforts |
| Refactoring existing service | 1 day |
| 2FA in user management service | 2 day |
| Fort password in user management service | 1 day |
| Login user validation API in user management service | 1 day |
| Protect user management service by token | 0.5 day |
| Client credential management in Auth service | 0.5 day |
| Read user info from token instead of payload in E-Wallet & QR service | 1 day |
| Validate Fruits vending clientId in QR service | 1 day |
| Add product management API in QR Service | 1 day |
| Add role restriction in QR Service | 1 day |
| Rewrite Web UI | 4 days |
| Android UI | 4 days |
| Add TLS | 1 day |
| Add data encryption method | 1 day |
| Code obfuscation | 1 day |
| Deploy app to Google play store | 2 day |
| Add Microsoft Intune plugin | 1 day |