

Project Report for Fruits Vending Membership

Practice Module for Certificate in Securing Ubiquitous Systems

**Team 2**

Members:

Xu Jiao(E0384412)

Zou Xuan (E0384799)

CONTENTS

1. Introduction 3

1.1 Project Context 3

1.2 Business Needs 3

1.3 Stakeholders 3

1.4 General Architecture 3

1.5 Project Scope 3

2. Project Conduct 4

2.1 Project Plan 4

2.2 Project Status 4

2.3 Project Metrics 4

3. Security Requirements 5

4. Threat Modelling 6

5. Security Architecture and Controls 7

6. Verification of Security Controls 8

7. <other things to be highlighted> 9

# Introduction

## Project Context

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.

## Business Needs

With this platform, the company will be able to recognize their customers and improve the quality of their business based the customer data/feedback that they will get from this platform.

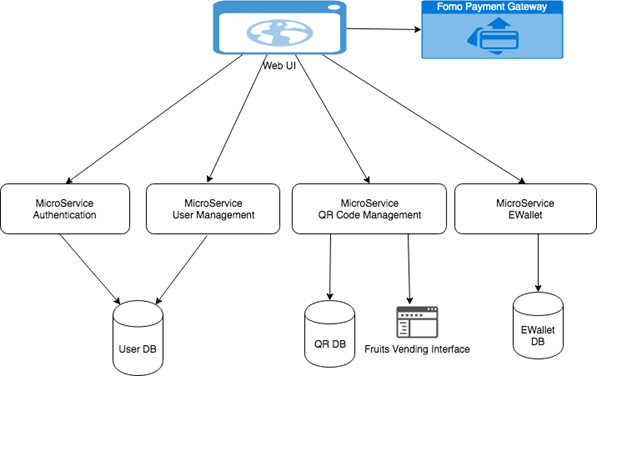
In addition to that, cloning/replicating the platform to other companies should be effortless so we can setup new business with other companies easily.

## Stakeholders

Fruits Vending Pte Ltd Business Team

Fruits Vending Pte Ltd Development Team

## General Architecture



## Project Scope

**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. (In Scope)

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. (Not in Scope)

**2) Authentication and Authorization**

a. Auth service is following OAuth2 password grant type. But currently is directly connecting User DB, which makes User DB has to open a firewall to Auth Service. This puts user data in risk if suth service is not handling data properly. Consider provide and API in user management service to validate user name and password, Auth service will call this API to validate user instead of directly connecting to DB.

(In Scope)

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. (In Scope)

c. The token issued does not include user’s role info. Will implement the role based access control for different type of user. (Not in Scope)

d. No revoke and refresh access token catered. (In Scope)

e. The token and configuration info was stored in memory, which made the service not able to scale horizontally. (In Scope)

**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 another 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.

(E-Wallet In Scope, QR out of Scope)

**4) Login with Facebook**

User had to sign up with a new account to login the system, with OpenID connect integration with the system, user could login with their Facebook ID. (In Scope)

# Project Conduct

## Project Plan

|  |  |
| --- | --- |
| Task | Efforts |
| 2FA in user management service | 2 days |
| Forget 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 | 0.5 day |
| Validate Fruits vending clientId token in QR service | 0.5 day |
| Add product management API / role segregation in QR Service | 1 day |
| Build web components | 3 days |
| integrate Mobile UI and Web UI with web components | 5 days |
| Add data encryption method/TLS2.0/http redirect to https | 1 day |
| Code obfuscation | 1 day |
| Deploy app to Google play store | 0.5 day |
| Add Microsoft Intune plugin | 0.5 day |
| Encrypt access token by session | 1 day |
| Implement email notification and add end to end encryption | 2 days |

## Project Status

* **The application is able to run as a native app in both android and iOS devices as well as a Progressive Web App.**
* **The application has below fully functioned features**

a. Register(with 2FA)

b. Forget password(with 2FA)

c. Login (with either Facebook account or signed up account within the application)

d. Top up E-Wallet through fomo payment gateway

e. E-Wallet transaction history search

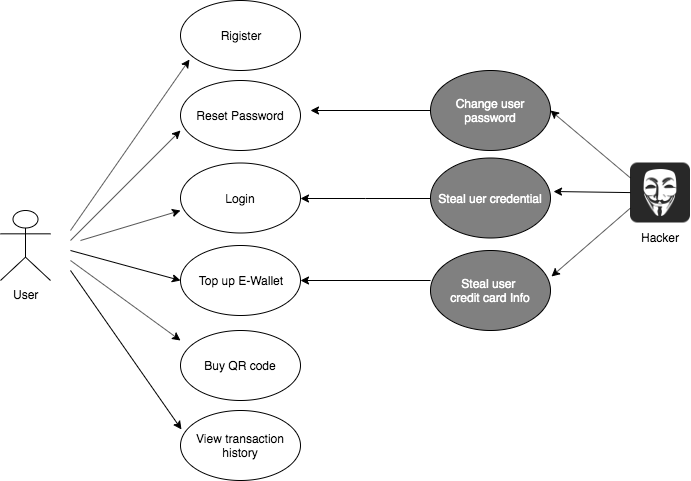
* **The QR management function is not fully integrated for time being, as the security issue has been addressed as POC in E-Wallet service.**
* **The role management is not implemented.**

## Project Metrics

|  |  |  |
| --- | --- | --- |
| Task | Efforts | Team member |
| 2FA in user management service | 2 days | Xu Jiao |
| Forget password in user management service | 1 day | Xu Jiao |
| Login user validation API in user management service | 1 day | Xu Jiao |
| Deploy app to Google play store | 0.5 day | Xu Jiao |
| Implement email notification | 2 days | Xu Jiao |
| Store configurable data and token in DB instead of in memory. | 1 day | Zou Xuan |
| Read user info from token instead of payload in E-Wallet service | 0.5 day | Zou Xuan |
| Build Mobile/ Web UI and integrate with backend services | 8 days | Zou Xuan 5 days/ Xu Jiao |
| Integrate login with Facebook (including exchange jwt token with Auth service) | 3 days | Zou Xuan |
| Refresh token, revoke token. Verify token directly through Auth service instead of locally and add cache mechanism. | 2 days | Zou Xuan |
| Add http redirect to https | 0.5 day | Zou Xuan |
| Deploy website to Google Firebase and other APIs to cloud | 1 day | Zou Xuan |
| Fortify SCA code scan and OWASP ZAP report | 0.5 day | Zou Xuan |

# Security Requirements

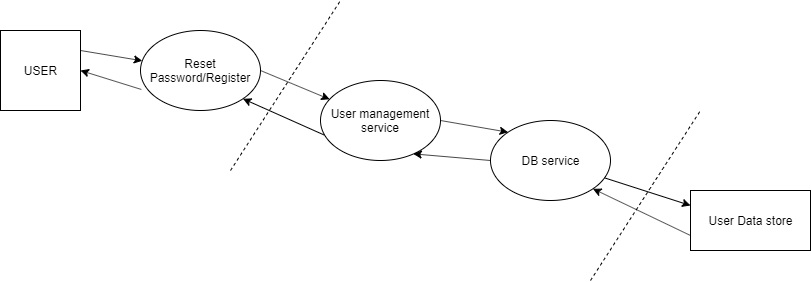
Identify and formulate the security requirements of the solution. The security requirements can be specified against groups of use cases or sets of features.

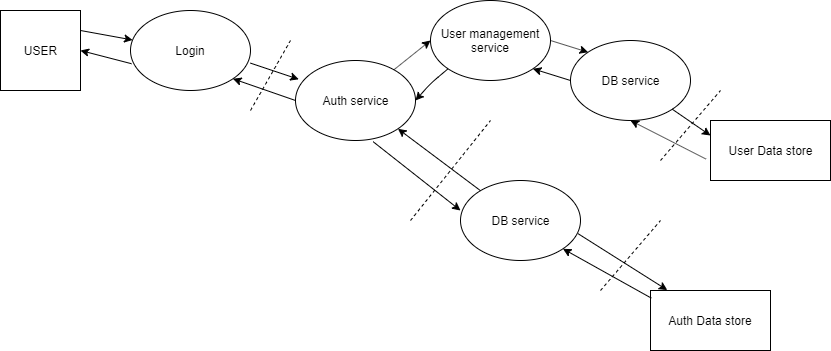


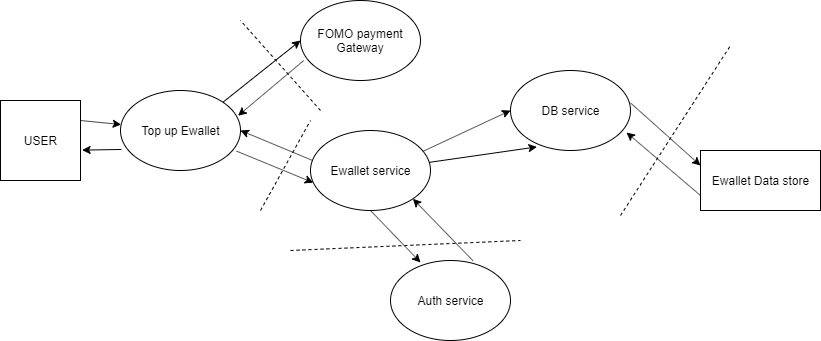
|  |  |
| --- | --- |
| **Requirement Category** | **Requirement description** |
| Authentication Requirements | Integrate Facebook with OpenID Connect protocol, 2FA for register  and reset password |
| Authorization Requirements | User is not allowed to access unauthorized page by directly copy url Backend services validate JWT token before returning response |
| Integrity Requirements | Generate and verify signature with apiKey for fomo payment |
| Availability Requirements | Deployed to cloud and enable auto scale & HA when traffic grows |
| Accountability Requirements | Add transaction history and server for traceability |
| Exception Handling Requirements | Provide general error message when exception occurred |
| Confidentiality Requirements | Store apiKey and client secrets in server environment variables without passing through internet All the services communicate through https |
| Session Management Requirements | Keep token timeout in 1 hour, auto refresh before expiry. Remove token from local storage  and revoke refresh token when logoff |

# Threat Modelling

Perform threat modelling to identify and quantify the threats associated with the solution. The threat model may include multiple levels of data flow diagrams. Where necessary, use suitable UML diagrams to illustrate detailed interactions between the architectural elements. Controls should be identified for the prioritised threats.







|  |  |  |  |
| --- | --- | --- | --- |
| **Data Flow Element** | **Threat Category** | **Identified Threat** | **Threat** |
| **All Use Case** | | | |
| User | Spoofing | Broken Authentication | Attacker impersonate other users to reset password/login. |
| Fomo payment gateway | Spoofing | Broken Authentication | Attacker impersonate other users to do payment. |
| All Use case data flow | Spoofing | Man-in-the-Middle | Attacker may alter the information to the server |
| All Service | Tempering | Cross Site Request forgery | Attacker may change the user information. |
| All Service | Elevation of privilege | Logic Flaw | Developer may use hardcode the program for testing which results in user making unauthorised request. |
| All Service | Spoofing | *Broken Authentication* | Attacker impersonate other users to top up E-Wallet. |
| All Services | Denial of services | Accepting Large Volume | Attacker supply high volume of data which server can’t handle the load. |
| All Services | Information Disclosure | Verbose Exception | Server response includes error information to user which reflects internal file structure or stack trace information. |
| All Services | Repudiation | Audit Log lacking | Attacker may modify user data without tracking |
| All Data Store | Repudiation | Audit Log Deletion | Attacker may use SQL statement to delete the log data store. |
| All Data Store | Tampering | SQL Injection | Attacker may use SQL statement in the input fields in order to do the damages on the database. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S/N** | **Identified Threat** | **Asset** | **Probability** | | | **Impact** | | **Risk**  **Level (P\*I)** |
| **DI** | **R** | **E** | **A** | **D** |
| 1 | Man-in-the-Middle | User Info,  E-Wallet Info,  QR Info | 2 | 3 | 1 | 3 | 2 | 30 |
| 2 | Verbose Exception | System Info, Database Info | 1 | 1 | 2 | 3 | 2 | 20 |
| 3 | SQL Injection | All Database Tables | 3 | 3 | 3 | 2 | 2 | 36 |
| 4 | Cross Site Request Forgery | User Credential,  E-Wallet balance | 2 | 2 | 3 | 2 | 2 | 28 |
| 5 | Audit Log Lacking | Traceability | 1 | 1 | 1 | 3 | 1 | 12 |
| 6 | Logic Flaw | System | 1 | 1 | 1 | 2 | 1 | 9 |
| 7 | Audit Log Deletion | Traceability | 3 | 0 | 0 | 1 | 1 | 6 |
| 8 | Broken Authentication | User Info,  E-Wallet Info,  QR Info | 2 | 3 | 3 | 2 | 2 | 32 |
| 9 | Accepting Large Volume | System Functionality | 3 | 2 | 2 | 3 | 1 | 28 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Identified Threat** | **Risk Level** | **Risk Ranking** | **Mitigation Control** | **System Traceability (Use Case)** |
| 3 | SQL Injection | 36 | 1 | Use ORM to access database | All |
| 8 | Broken Authentication | 32 | 2 | Hashing with salt for password mechanism. Use jwt token to protect backend services. | Login, top up E-wallet, Buy QR code. |
| 1 | Man-in-the-Middle | 30 | 3 | Use HTTPS connection with TLS 1.2 | All |
| 9 | Accept-Large-Volume | 28 | 4 | Use client side input validation. Use maxAllowed content length. Use excution timeout. | All |
| 4 | Cross Site Request Forgery | 28 | 4 | Referrer cross origin check | All |
| 2 | Verbose Exception | 20 | 5 | Use custom error message  Adopt the good practice of error  exception handling | All |

# Security Architecture and Controls

Highlight the key design security decisions. Prescribe adequate security controls to address the vulnerabilities of the solution. Illustrate the resultant general architecture with security controls.

# Verification of Security Controls

Describe the test cases and the results of verification of the security controls. Highlight the issues found and their corrective actions.

|  |  |
| --- | --- |
| Security Controls | Verification |
| SQL Injection | Cannot login system with customised sql |
| Broken Authentication | Put invalid token in Swagger page |
| Man-in-the-middle | 1.Website and backend service URL can be accessed by https only  2.SSL/TLS protocol versions |
| Accepting Large Volume | UI Validation on long data |
| Cross Site Request Forgery | Cannot submit data through customised html |
| Verbose Exception | Wrong Password error message |

# <other things to be highlighted>

ZAP Report

Fortify SCA report

SSL Report