FYP 2022-2023

Project Proposal

Ethical Hack Bot

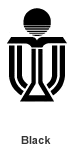
Group 2

PriceWaterhouseCoopers

Prepared By

Leung Ka Ho [khleungaw@connect.ust.hk]

Chan Yu Hin [yhchanbd@connect.ust.hk]



CONTENTS

COVER PAGE 1

TABLE OF CONTENT 2 – 3

PROJECT DESCRIPTION

Introduction 4

Company Background 4

Description of Current System 4

Problems Statements 5

Objectives 5

Project Scope and/or constraints 5

BACKGROUND STUDY - RESEARCH

Information about Related Systems and Applications 6

System Architectures 6-7

System Features 7

System Requirements 8

Sources 8

Comparisons 9

Summary 9

SYSTEM

OAuth System Specifications and Requirements 10

OAuth System Features 10

OAuth System Architecture 11

Jira System Specifications and Requirements 12

Jira System Features 10

Jira System Architecture 13

ANALYSIS OF USER REQUIREMENTS

Sole Data Source and Conclusion 14

TECHNOLOGY

Evaluation of Technology to be adopted 15

Operating Systems 16

Web Server 17

SCHEDULE 18-23

TASK DIVISIONS 24-26

REFERENCES 27-28

APPENDIX

Agenda and Minutes of all Meetings 1

**Project Description**

**Introduction**

This project aims to improve the user experience and functionality of the existing Hack Bot, a threat and vulnerability management tool, according to the client request. Two features, single sign-on with OAuth and Jira integration, will be intergrated to Ethical Hack Bot through this project.

**Company Background**

PrincewaterhouseCoopers (PwC) is an international brand of firms offering various professional services, such as auditing, consulting, and risk assurance. Within the domain of Information Technology, PwC provides solutions to a range of issues, including cybersecurity, data analytics, and IT transformation. Our project is supervised by Dark Lab under PwC which focuses on cybersecurity. Dark Lab helps companies design cybersecurity solutions as well as assist in the implementation. Also, Dark Lab has developed digital products such as Ethical Hack Bot, which provides threat and vulnerability management automation. ((PricewaterhouseCoopers)

**Description of Current System**

Ethical Hack Bot is a platform that allows the automation of various cybersecurity management processes. The system covers asset management, scanning task management, report management, vulnerability management and attack surface management. The system integrates solutions related to these aspects in a single platform, allowing automation and control of workflow with increased efficiency and speed. (PricewaterhouseCoopers)

**Problems Statements**

1. An extra set of usernames and passwords need to be managed for system access.
2. Findings from vulnerability scans need to be reported to users promptly and efficiently.

**Objectives**

1. Implement single sign-on using OAuth. Allow users to access the system by logging in through Microsoft AD or Okta.
2. Integrate with Jira so that findings create corresponding issues on Jira when findings are available.

**Project Scope and/or constraints**

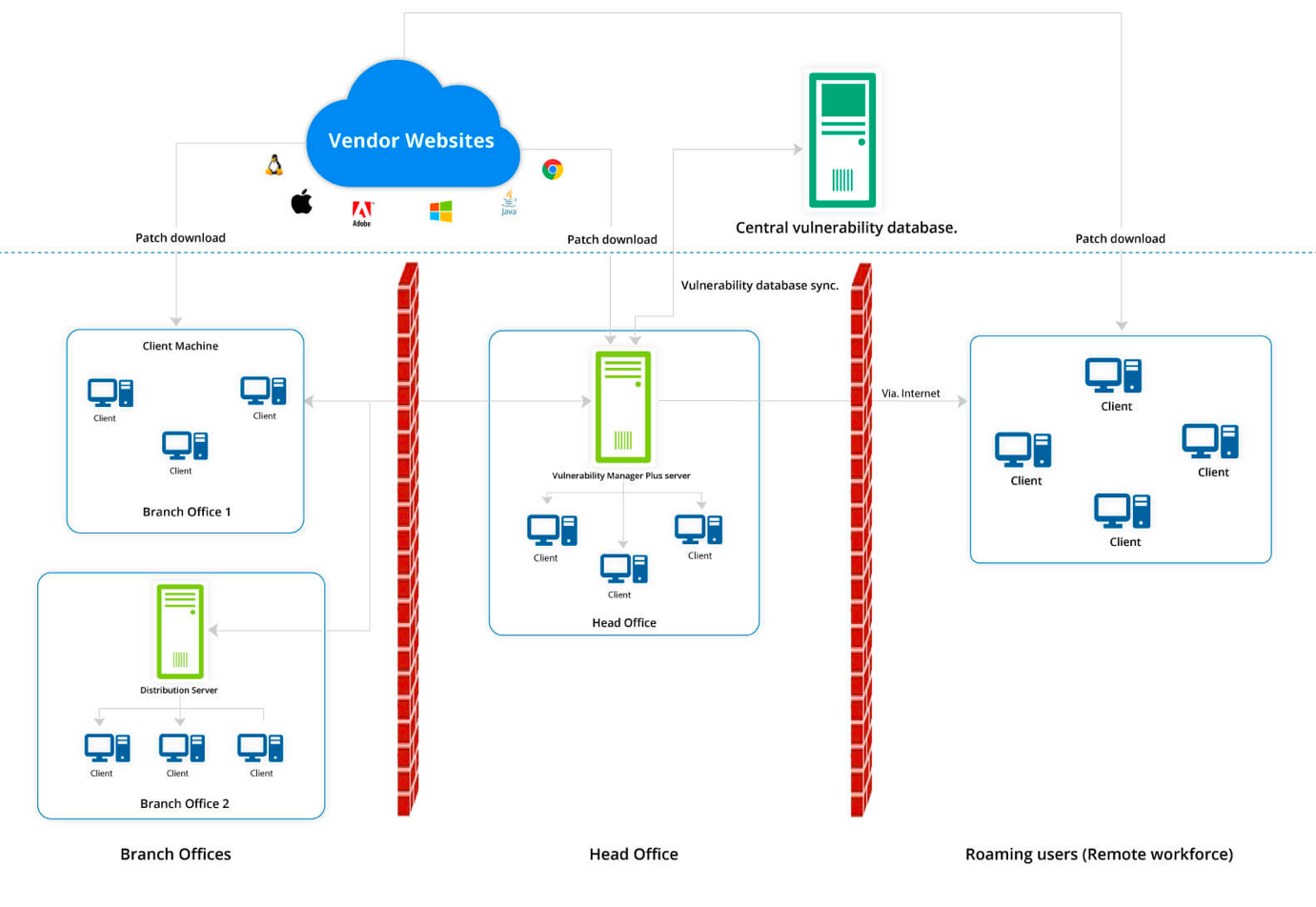
Production-ready level of Jira integration and OAuth integration.

**Background Study - Research**

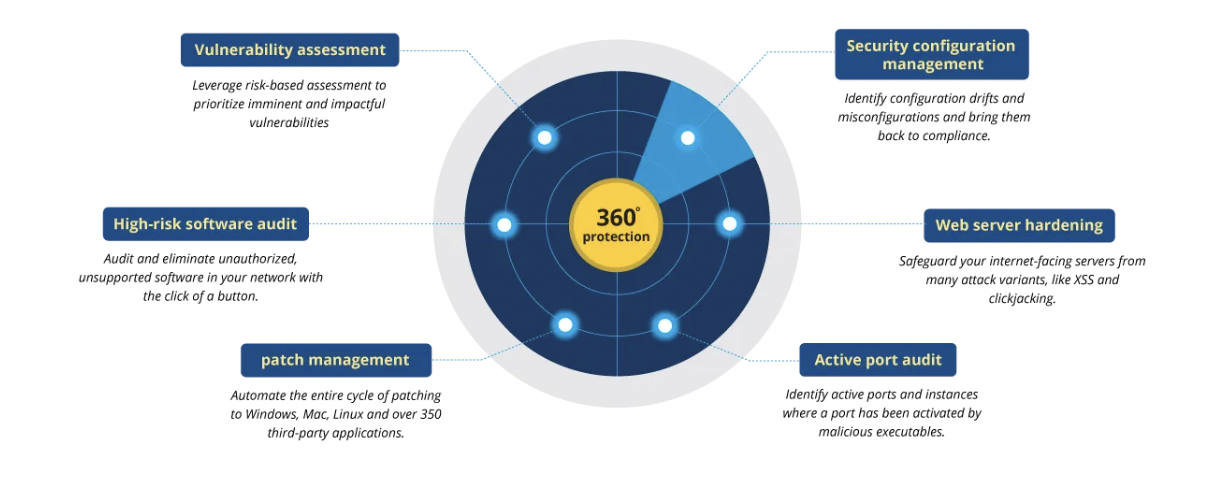
**Information About Related Systems and Applications**

Hack Bot is identified as a vulnerability management platform. Vulnerability management is a cyclical process of identifying IT assets and correlating them with a continually updated vulnerability database to identify threats, misconfigurations, and vulnerabilities. Another aspect of vulnerability management including validating the urgency and impact of each vulnerability based on various risk factors and responding to the critical threats swiftly. (David Bisson, n.d.)

**System Architectures**

A general vulnerability management platform works on a client-server architecture. and contains 2 components: vulnerability management system and IT assets including computer devices, IP address, file system etc. The centralized vulnerability management system in charge of assigning and organising vulnerability scanning, filtering, and storage to the database. The database would keep track of the latest scan and remediation information. Most existing vulnerability management support multiple third-party scanning tools, which scanning task would be send to those cloud-based tools as a task and received scanned result for further processing.

**System Features**

Highlighted features of vulnerability assessment include:

1. SQL Injections
2. Cross-site Scripting
3. Broken Authentications
4. web server configuration and Malware issue
5. IP Range and active port scanning

Additional features that improve user experiences include:

1. Issues exporting to third party application
2. Reporting
3. Visualization

(*Vulnerability management, 2022)*

**System Requirement**

The overall requirements of a TVM system can be divided into 4 core steps with robotic automation: identify, evaluate, treating and reporting.

* 1. Identify vulnerability along client’s context, including network-accessible system, ports, database, servers, enterprise asset including files and system configuration.
  2. Evaluate Vulnerability by filtering false positive cases, the risk posed by it and their risk ratings, such as the Common Vulnerability Scoring System, to help better prioritise issues and address them appropriately in accordance with an organisation’s risk management strategy.
  3. After validation of a vulnerability, there are different ways for treatment, including:
  4. Remediation: fully fixing or patching to prevent exploitation of a vulnerability
  5. Mitigation: lessening the likelihood or impact of the vulnerability being exploited
  6. Acceptance: taking no action to fix for low-risk vulnerability
  7. Visualize and report scanning result by risk level in a timely and organised format
  8. Automate the whole process with the minimum manual effort involved

(Vulnerability Management process and Tools, 2022)

**Sources**

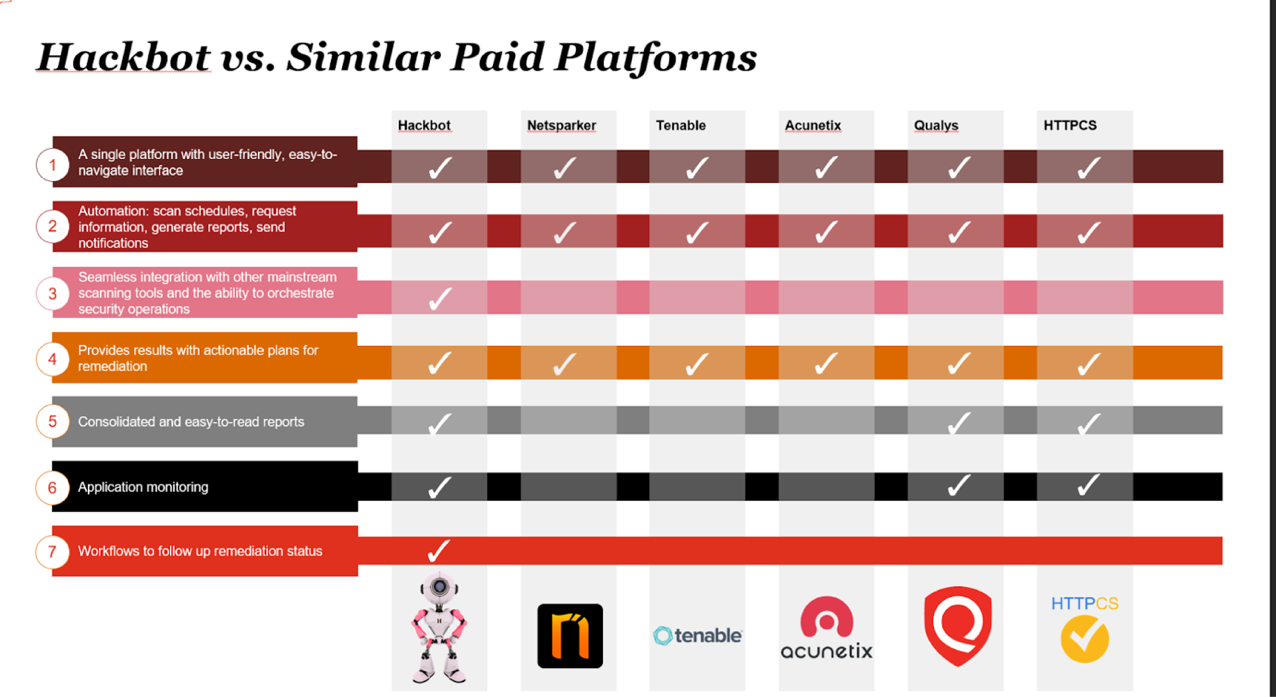
David Bisson. (2018). *What is Vulnerability Management Anyway?* Tripwire. Retrieved October 20, 2022, from https://www.tripwire.com/state-of-security/what-is-vulnerability-management-anyway

*Vulnerability management: Definition and process*. What is vulnerability management? | Vulnerability management process - ManageEngine Vulnerability Manager Plus (2022). Retrieved October 20, 2022, from https://www.manageengine.com/vulnerability-management/what-is-vulnerability-management.html

*Vulnerability Management process and Tools*. Snyk. (2022, May 19). Retrieved October 20, 2022, from https://snyk.io/learn/vulnerability-management/

**Comparisons**

Provided by Tony, below is the competitor analysis of Hack Bot.



Some renowned cybersecurity management platforms include:

1. Acunetix by Invicti Security, which specialises in automatic website security scanning, penetration testing and external vulnerability scanning targeting SMEs applications. Given its scanning engine is written in C++, It's known for its fast performance (About Acunetix, n.d.)
2. Qualys VMDR: Qualys was the 1st SaaS TVM platform launched in 1999. Its Vulnerability Management Detection and Response system is accessed as a cloud service with a voice agent, virtual scanner and passive network scanning capabilities to provide asset management and scanning service. (Qualys VMDR 2.0, n.d.)

**Summary**

To conclude, we can observe that while most existing TVM platforms provide similar features and capabilities in security scanning and detection, Hack Bot outperforms its competitors in terms of integration with other scanning tools and the ability to orchestrate security operations as in the automation aspect.

**System**

**Specific System Specifications and Requirements**

The whole project mainly contains 2 systems: OAuth System and Jira Issue Management System.

1. **OAuth System**

**Functional Requirement**

Allow users to log in using single sign-on through Microsoft AD or Okta

On the login page, an option for SSO will be available. Upon choosing to login using SSO, the user will be redirected to the authentication server of AD or Okta and complete login on AD or Okta. Afterwards, the user will then be redirected back to Hack Bot and be able to access Hack Bot.

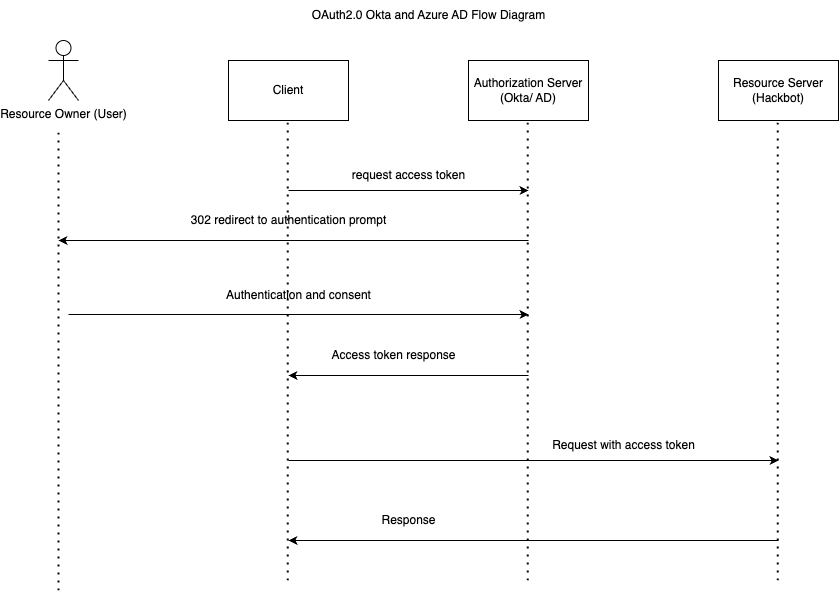
**Non-functional Requirement**

* + Integrate with the existing logic for login and access control such that user will have access corresponding to their roles
  + Ensure that the login process is secure

**System Features**

Provide option to access the system through SSO to users.

**System Architecture**



* 1. Client request authorization from the resource owner (user)
  2. If the user gives authorization, the client passes the authorization grant to the authorization server (AD / Okta)
  3. If the grant is valid, the authorization server returns an access token, possibly alongside a refresh and/or ID token.
  4. The client now uses that access token to access the resource server.
  5. The new OAuth system would map the client’s role on the token to the existing role-based access control system for resource accessing.

1. **Jira Issue Management System**  
   Given the existing Hack Bot provided a real time dashboard for scanning result and remediation visualization, some of its client prefers to use their existing Jira dashboard for managing cybersecurity issue. Thus, on one side existing scanning result on Hack Bot need to be able to export to Jira. On the other side, for any future scanning result, it should also be capable of exporting to Jira. And the whole exporting process should be automated.

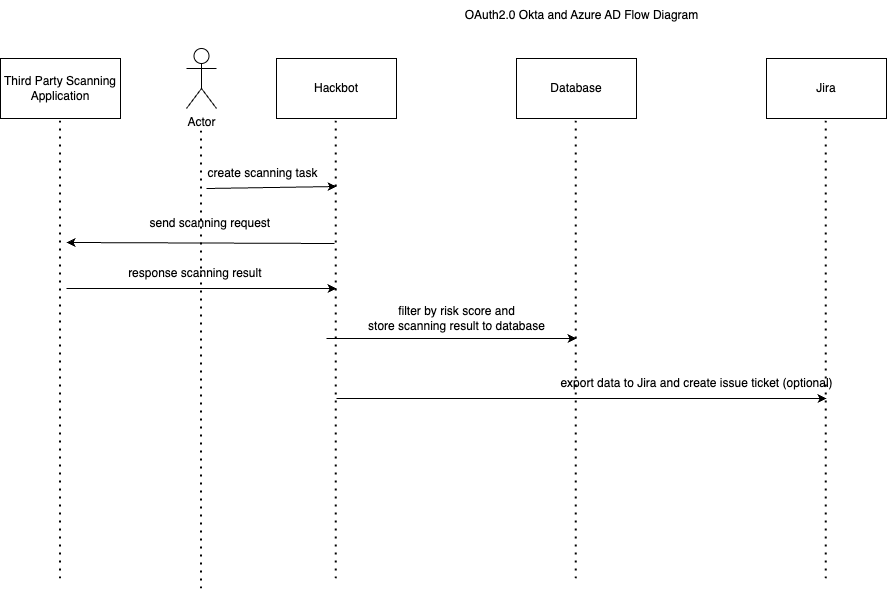
**Functional Requirement**

* + Generate issues on Jira when there are new findings
  + When new findings are available after scanning tasks, issues should be generated on Jira. The issues should contain information from the findings. Related attachments should also be included in the issues.
  + Update issues when new findings are detected on existing issue

**System Features**

Allow users to track findings as issues on Jira.

**System Architecture**



* 1. New scanning result generated by the 3rd party application and send to Hack Bot
  2. Hack Bot stores filter result by risk level and store to database
  3. Hack Bot export data to Jira and create issue ticket

**Analysis of User Requirement**

The user requirements are provided directly by the project supervisor.

The clients of Hack Bot requested single sign-on through Microsoft AD or Okta as they have been using these services and would prefer to use existing account instead of managing more accounts. Meanwhile, their clients would also like to integrate with Jira so that findings that require action would be listed on Jira as issues.

Considering that our user requirements have been defined clearly by our project supervisor in accordance with their client, further analysis is not needed in this project.

**Technology**

**Evaluation of Technology to be adopted**

We have gathered a brief picture of the tech stack that we are applying for this project from Tony. Here is a summary of them.

|  |  |  |
| --- | --- | --- |
| Technology |  | Do students have the capability/ relevant experiences? |
| Operating System | Linux | Yes |
| Programming Language | Node.js | Yes |
|  | JavaScript Axios Library | Yes |
|  | Vue.js | No |
|  | Vue Passport | No |
| Database Management System | MongoDB | Yes |
| Web Server/ Application Server | Express | Yes |

**Operating Systems**  
The hack bot is developed and hosted on Linux OS. Linux is a Unix-like, open source and community-developed operating system (OS) for computers, servers, mainframes, mobile devices and embedded devices. It is supported on almost every major computer platform, including x86, ARM and SPARC, making it one of the most widely supported operating systems.

Yet, we are not involved in any web host/ environment configuration for this project. Operating System is not a key technology here. (What is linux?, 2022)

**Programming Languages and Libraries**

Node.js  
For the backend, including data communication through REST API and internal logic. The current project is using version 14.20.1 Node.js.  
Node.js is a server-side platform based on the JavaScript Engine in Google Chrome. Node.js is an open-sourced cross-platform runtime environment for developing server-side and networking applications. Node.js also comes with a big library of JavaScript modules, which makes developing Node.js web applications much easier. (Node.js, n.d.)

Axios  
For making Rest API with third-party applications like Jira, we are using the Axios library to make the rest API calls to their endpoints.  
Axios is an HTTP client library based on promises. It makes sending asynchronous HTTP requests to REST endpoints easier and helps you perform CRUD operations. Compared with the standard fetch function in JavaScript, Axios signifies the data automatically with better error handling. (Getting started | Axios Docs, n.d.)

Vue.js  
For the frontend, Vue.js is adopted for building web interfaces.  
Vue.js is an open-sourced JavaScript framework for building user interfaces (UIs) and single-page applications (SPAs). It utilises a model-view-viewmodel (MVVM) architectural pattern. (Vue.js, n.d.) **Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped into any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more. (Vue-Passport, n.d.)**

**Database Management System**

MongoDB

The Hack Bot uses MongoDB to store security events logs and issues details. Possible usage for this project includes retrieving security issue data and exporting to Jira through Jira’s developer API.  
MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. (The developer Data Platform. MongoDB, n.d.)

**Web Server**

Node.js with Express

The Hack Bot web application used Express for the web server.

Express.js is a framework based on Node.js, which is used for building web-application using approaches and principles of Node.js.event-driven. (Node.js web application framework. Express, n.d.)

**Schedule**

Our internal working mechanism is that we would meet up every week for half an hour to one hour to update our works and plan ahead to ensure we can fulfil our target deliverables on time.

External wise, we have built a consensus with Wilson to meet up bi-weekly to update our progress and share any difficulties.

We will also chat via WhatsApp(internal) and Slack(external) to maintain a frequent and efficient communication on the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Date | Internal | School | External |
| 1 (1-3/9/2023) | 31/8/2022 |  | Announcement of Projects |  |
|  | 3/9/2022 | Discuss project preference, ice breaking |  |  |
| 2  (4-10/9/2023) | 4/10/2022 | Contact and Schedule meeting with PwC |  |  |
|  | 7/9/2022 | submit project choice to TA |  | Kickstarter meeting with PwC |
|  | 8/9/2022 |  | Deadline of project selection |  |
| 3  (11-17/9/2023) |  |  |  |  |
| 4 (18-24/9/2023) |  |  |  |  |
| 5 (25/9-1/10/2023) | 27/9/2022 |  |  | 2nd Meeting (Go through product demo and project requirements of Jira Module) |
| 6 (2-8/10/2022) |  |  |  |  |
| 7  (9-15/10//2022) | 13/10/2022 |  |  | 3rd Meeting (Set up development environment) |
| 8  (16-22/10/2022) | 20/10/2022 | finish proposal |  |  |
|  | 21/10/2022 |  | Proposal Submission |  |
| 9 (23-29/10/2022) |  | Finish Jira API study and built proof of concept |  | Planned 4th Meeting(TBC) |
| 10 (30/10-5/11/2022) |  | Schedule meeting with TA to review proposal work |  |  |
| 11 (6-12/11/2022) |  | Finish studying existing Vue and Node.js codebase |  | Planned 5th Meeting(TBC) |
| 12 (13-19/11/2022) |  | Schedule meeting with Prof to discuss midterm deliverables |  |  |
| 13 (20-26/11/2022) | 23/11/2022 | Finish midterm report |  |  |
|  | 25/11/2022 |  | Midterm report submission |  |
| 14 (27/11-3/12/2022) |  |  |  | Planned 6th Meeting(TBC) |
|  | 3/8/2022 | Finish midterm presentation material |  |  |
| 15 (4-10/12/2022) | 5-8/12/2022 |  | Midterm Presentation |  |
| 16 (11-17/12/2022) | 11-17/12/2022 | Final Exam Break | | |
| 17 (18-24/12/2022) | 19/12/2022 |  | End of 2022 Fall |  |
| 18 (25-31/12/2022) |  | Christmas Break | | |
| 19 (1-7/1/2023) |  |  |  | Planned 7th Meeting(TBC), Finish Jira Module |
| 20  (8-14/1/2023) |  |  |  |  |
| 21 (15-21/1/2023) |  |  |  |  |
| 22 (22-28/1/2023) | 22-29/1/2023 | Chinese New Year Break | | |
| 23 (29/1-4/2/2023) | 3/2/2023 | Finish Okta and AD proof of concept with vue js and API study | Commencement of 2023 Spring | Planned 8th Meeting(TBC), |
| 24 (5-11/2/2023) |  |  |  |  |
| 25 (12-18/2/2023) |  |  |  | Planned 9th Meeting(TBC), |
| 26 (19-25/2/2023) |  |  |  |  |
| 27 (26/2-4/3/2023) |  | Finish half of the module (Okta/ AD depends on Wilson choice) |  | Planned 10th Meeting(TBC), |
| 28 (5-11/3/2023) |  |  |  |  |
| 29 (12-18/3/2023) |  |  |  |  |
| 30 (19-25/3/2023) |  |  |  | Wrap up meeting (TBC) |
|  | 24/3/2023 |  |  | Finish OAuth Module |
| 31 (26/3-1/4/2023) |  | Schedule meeting with Prof and TA to discuss final deliverables. |  | Spared Meeting Week |
| 32 (2-8/4/2023) | 2-8/4/2023 | Midterm break | | |
| 33 (9-15/4/2023) | 9-11/4/2023 | Midterm break | | |
| 34 (16-22/4/2023) | 20/4/2022 | Finish final report |  |  |
| 35 (23-29/4/2023) | 28/4/2023 |  | Final Report Submission |  |
| 36 (30/4-6/5/2023) | 30/4/2022 | Finish final presentation preparation |  |  |
|  | 2-5/5/2023 |  | Final Presentation |  |

**Task Division**

Tasks under this project's scope include 3 main categories: communication, documentation, and development. Below is a summary of our task division.

|  |  |  |
| --- | --- | --- |
| Task |  | Person In Charge |
| Communication | External Communication | Allen |
|  | Internal Communication | Daniel |
| Documentation | Technical Documents | Allen |
|  | Supplementary Documents | Daniel |
| Development | Frontend (Vue.js) | Allen |
|  | Backend (Node.js & Express) | Daniel |

**Development**

Frontend: refers to any development process related to the web interface and Vue.js, including reading existing Vue.js codebase, study vue passport library for Okta and AD authentication and authorisation.

Backend: refers to any development process related to the database and logical operation of Hack Bot and Node.js, including reading existing API codebase, studying Jira developer API and access control/ identity management component of OAuth.

**Communication**  
External communication: refers to the communication between students and the PwC project supervisors. Possible sub-tasks include scheduling kickstarting and bi-weekly catch-up meetings, gathering and discussing project requirements, and setting up the development environment. Allen will be in charge of this task by maintaining a close connection with Tony and Wilson to ensure our deliverables match their expectations and requirements.

Internal communication: refers to the communication between students, course instructors, and TAs. Possible sub-task includes scheduling catch-up meeting to discuss course requirement for proposal, mid-term and final presentation and reports, or any difficulties related to the project. Daniel will be in charge of this task to ensure proper documentation and presentation is prepared for the course.

**Documentation**

Technical Documentation: content includes architecture, requirement analysis and research findings of Hack Bot System’s, project requirements and third-party applications like Jira, Okta and OAuth, Allen will be in charge of this task.

Supplementary Documentation: content includes drafting minutes, the agenda of every meeting, and company research. Daniel will oversee this task. Minutes would be complete on the week of the meeting. Yet, consider the minutes documentation is for school grading purpose, we would be sending them to project supervisor for approval in batch to minimize their workload.

**References**

*About Acunetix, developer of the leading web vulnerability scanner*. Acunetix. (2021, June 23). Retrieved October 20, 2022, from https://www.acunetix.com/about/

Atlassian. (n.d.). *Jira software for developers: A guide to getting started*. Atlassian. Retrieved October 20, 2022, from https://www.atlassian.com/software/jira/guides/developers/basics

Barclayn. (n.d.). *What is Azure Active Directory? - microsoft entra*. Microsoft Entra | Microsoft Learn. Retrieved October 20, 2022, from https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis

*The developer Data Platform*. MongoDB. (n.d.). Retrieved October 20, 2022, from https://www.mongodb.com/

*Getting started*. Getting Started | Axios Docs. (n.d.). Retrieved October 20, 2022, from https://axios-http.com/docs/intro

*Implement oauth for Okta*. Implement OAuth for Okta | Okta Developer. (n.d.). Retrieved October 20, 2022, from https://developer.okta.com/docs/guides/implement-oauth-for-okta/main/

Janicericketts. (n.d.). *OAUTH 2.0 authentication with Azure Active Directory - Microsoft entra*. OAUTH 2.0 authentication with Azure Active Directory - Microsoft Entra | Microsoft Learn. Retrieved October 20, 2022, from https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/auth-oauth2

*Jira rest api examples*. Jira REST API examples. (n.d.). Retrieved October 20, 2022, from https://developer.atlassian.com/server/jira/platform/jira-rest-api-examples/

Mattwojo. (n.d.). *NodeJS on windows*. NodeJS on Windows | Microsoft Learn. Retrieved October 20, 2022, from https://learn.microsoft.com/en-us/windows/dev-environment/javascript/nodejs-overview

*Node.js web application framework*. Express. (n.d.). Retrieved October 20, 2022, from http://expressjs.com/

Node.js. (n.d.). Node.js. Retrieved October 20, 2022, from https://nodejs.org/

*OAuth 2.0 and openid Connect Overview*. OAuth 2.0 and OpenID Connect Overview | Okta Developer. (n.d.). Retrieved October 20, 2022, from https://developer.okta.com/docs/concepts/oauth-openid/

PricewaterhouseCoopers. (n.d.). *Dark lab*. PwC. Retrieved October 20, 2022, from https://www.pwchk.com/en/issues/cybersecurity-and-privacy/dark-lab.html

PricewaterhouseCoopers. (n.d.). *Ethical Hack Bot: Your cybersecurity guardian*. PwC. Retrieved October 20, 2022, from https://www.pwchk.com/en/issues/cybersecurity-and-privacy/ethical-hack-bot.html

PricewaterhouseCoopers. (n.d.). *US PWC*. PwC. Retrieved October 20, 2022, from https://www.pwc.com/

*Qualys VMDR 2.0: - vulnerability management tool*. Qualys. (n.d.). Retrieved October 20, 2022, from https://www.qualys.com/apps/vulnerability-management-detection-response/

*Vue-Passport*. npm. (n.d.). Retrieved October 20, 2022, from https://www.npmjs.com/package/vue-passport

Vue.js - The Progressive JavaScript Framework | Vue.js. (n.d.). Retrieved October 20, 2022, from https://vuejs.org/

*What is linux?* Linux.com. (n.d.). Retrieved October 20, 2022, from https://www.linux.com/what-is-linux/

**Appendix**

**Agenda and Minutes of all Meetings**

************

****