

AITA: Asset and Inventory Management System for Research Grants

Brynard Bryan Edmund and Shahida Sulaiman
Faculty of Computing
Universiti Teknologi Malaysia
81310, Johor Bahru, Johor, Malaysia
bryanedmund@graduate.utm.my, shahidasulaiman@utm.my

Abstract—Research Management Center (RMC) at Universiti Teknologi Malaysia (UTM) is a research management center that has been actively involved in managing and monitoring research activities at the university. However, the current manual recording system using KEW.PA forms for asset and inventory management is inefficient. Moreover, the lack of a platform for transaction history and the challenge of updating the value of assets and inventories pose additional problems. To address these issues, this work introduces AITA, a Web-based application that offers a centralized system for recording items and streamlines loan management for easier transactions among projects. The implementation of AITA successfully achieves its development objectives, providing a reliable solution for asset and inventory management needs at RMC.

Keywords—Web application; asset and inventory management; university; research grant

I. INTRODUCTION

Inventory management involves methods used to control and organize goods or materials in a facility, allowing companies and organizations to operate efficiently and cost-effectively. According to Vinculum [1], there are five benefits to inventory management: centralized inventory monitoring, expense control, improved transactions, managed planning and forecasting, and reduced time for inventory management.

Jenkins [2] identifies three types of inventory management systems: manual inventory systems, periodic inventory systems, and perpetual inventory systems. Manual systems rely on paper records or spreadsheets, periodic systems involve manual and periodic counts with a database for record-keeping, and perpetual systems provide real-time stock data using technologies like RFID tags.

As project leader in RMC UTM, an organization focused on research activities and grants, there are several problems with their current inventory management. After receiving funds or grants, project leaders purchase assets or inventory items, recording them through KEW-PA forms and submitted to multiple departments. This lack of centralized data leads to synchronization issues and the potential loss of forms. Additionally, there is no platform for transaction history,

making it difficult for project leaders to track the availability of items. The records of asset and inventory values are also challenging to maintain, with the accumulation of KEW.PA forms affecting cost tracking.

The aim of this project is to develop an asset and inventory management system known as AITA, specifically for managing research grants, addressing their current problems and facilitating the management of assets and inventory. AITA will allow centralized tracking of inventories and assets, enabling project leaders to add, edit, or delete items. It will also provide user authorization for security purposes, ensuring access is limited to authorized personnel. Furthermore, AITA will support loan management, indicating the availability of items to other project leaders. The objectives of AITA are to elicit requirements, design and develop the application, and test it using suitable techniques. The scope of AITA includes its use on browser with network connectivity. The inventory and asset management usage restricted to project leader and the organization's team. AITA is of great importance to project leaders as it resolves many of their existing problems. It provides a centralized platform for tracking inventories and assets, enhances efficiency in recording project items, and offers user authorization for data security. Additionally, AITA facilitates loan management, ensuring project leaders can easily check item availability.

The following sections include Section 2 on the literature review, Section 3 elaborates the methodology, Section 4 states the analysis and design, followed by Section 5 on implementation and testing. Section 6 concludes the project and its possible future works or enhancements.

II. LITERATURE REVIEW

There are a lot of item tracking applications in the world. For the project references, there are several sites included in this study to discover new potential features to be added to AITA. This section will further detail the analysis of existing applications which are Sortly [3], GoCodes [4], Stock and Inventory Simple (SaIS) [5], and UTMFin [6] as compared to the proposed AITA.

Sortly [3] is a highly rated inventory tracking app available for free on Android and iOS. It offers features for item management, barcode connectivity, and compatibility with handheld scanners. Users can log in using various methods like Google, Apple ID, or SSO, and there is a “forgot password” function for password recovery. The dashboard provides an inventory summary, recent activity, and a sidebar for item management, searching, and report generation. Users can input item details, photos, and optional fields like tags and notes, which can be edited later.

GoCodes [4] is a subscription-based inventory tracking app designed for industry customers. It offers features to efficiently manage company inventory, including asset IDs and different viewing options such as an asset list, map, and pictures. Users can fill in asset data based on their company's requirements, with no required fields.

SaIS [5] is a mobile-based inventory tracking app suitable for small businesses and home users. It helps users track inventory, sales, and purchases, and offers additional features like barcode scanning, importing and exporting Excel files. The main page allows users to perform various functions, such as adding items, viewing transaction history, generating reports, and managing expenses. It also includes a contact management function for customers and suppliers.

UTMFin [6] is an existing inventory tracking app used by RMCUTM, but its shortcomings and lack of practicality have been noted by stakeholders. Limited access to the system has led to most research being based on the manual book. The dashboard of UTMFin allows users to perform functions like accepting assets and managing them. It includes fields for the project leader's name, item description, loan dates, and return dates. The loan function requires the project leader to press “sah”.

SaIS [5] is the only inventory management application among Sortly [3], GoCodes [4] that does not allow users to record the value of assets in the system. Both Sortly [3] and GoCodes [4] offer this functionality, enabling users to track the cost of items. When comparing the features of these applications, Sortly [3] stands out as a comprehensive inventory tracking solution. In addition to its basic inventory recording capabilities and provides user authentication for enhanced security, a transaction history to review recent activities, the ability to generate inventory reports, and the option to organize inventory into multiple folders for better sorting.

On the other hand, AITA meets all the requirements specified by RMC UTM. Since there is no other application that addresses all of RMC UTM's challenges, AITA emerges as the best choice for their needs. In addition to inventory recording, AITA offers features for managing loans, allowing users to track which individuals currently have borrowed assets or inventory. It also generates an inventory summary to provide an overview of the project's assets and inventory. Lastly, AITA incorporates user authentication to ensure security.

Laravel is a widely used PHP framework for web development, following the Model-View-Controller (MVC) architectural pattern. It offers advantages such as separation of concerns and specialization for developers. With its rich

features and tools, including routing, query building, authentication, and caching, Laravel provides a robust platform for web application development.

In the context of the AITA project, the decision was made to switch from the initially planned Spring framework to Laravel. This choice was driven by several factors. Firstly, Laravel's PHP-based nature aligns with the development team's expertise and familiarity, reducing the learning curve and increasing productivity. Secondly, Laravel's emphasis on rapid development, along with its built-in features and well-defined structure, allows for efficient web application development. The extensive Laravel ecosystem with its packages and libraries further accelerates the development process. Considering these factors, including the team's familiarity with Laravel and its focus on rapid development, Laravel was chosen as the preferred framework for the AITA project

III. METHODOLOGY

Agile methodology divides a project into phases with continuous engagement among collaborators and stakeholders. It involves planning, implementation, and reviewing cycles to deliver high-quality software [7]. For AITA, agile methodology is the ideal approach, allowing developers to adapt processes based on stakeholder requirements. Since AITA is specifically designed for RMC UTM's internal use, tailoring the software to fit their business processes is crucial. Agile methodology facilitates an easier development process for AITA.

Among various agile methodologies, AITA utilizes the Kanban framework. Kanban enables visualizing the workflow and provides clear communication between RMC stakeholders and developers. It allows stakeholders, both internal and external, to understand the system's operation. According to Wrike, Kanban's core principles include visualizing and actively managing the workflow, creating process guidelines, using feedback loops, and promoting evolution. Implementing Kanban involves using a Kanban board to map out the workflow and track task progress. The board typically consists of three columns: To-Do, On-Going, and Completed, with tasks represented by cards. These cards contain information such as descriptions, assignees, estimated completion time, and requirements for progressing to the next stage [7].

To develop and run AITA, specific software and hardware requirements must be considered as shown in Table I. Among the software used include Lucidchart [8] for diagrams and Spring Boot [9] for the interface. In addition, the Integrated Development Environment (IDE) chosen was VS Code [10] and MySQL [11] for the database.

TABLE I. SOFTWARE AND HARDWARE REQUIREMENT

Software	Description
Operating System	Window 10 Home Single Language
Diagram	LucidChart
Documentation	Microsoft Office Word
Coding development	Netbeans
Database System	MySQL
Presentation	Microsoft Office Powerpoint

Hardware	Requirement
Processor CPU	Intel Core i7-8750H Processor CPU @ 2.20 GHZ
Memory	4 GB RAM
Data and Bandwidth	3Mbps
Drive	HDD or SSD

IV. ANALYSIS AND DESIGN

After studying the existing solutions, the requirements were gathered from the stakeholders to ensure IATA meets the specific requirements for asset and inventory management at RMC, UTM. Figure 1 shows seven use cases to represent the high-level requirements.

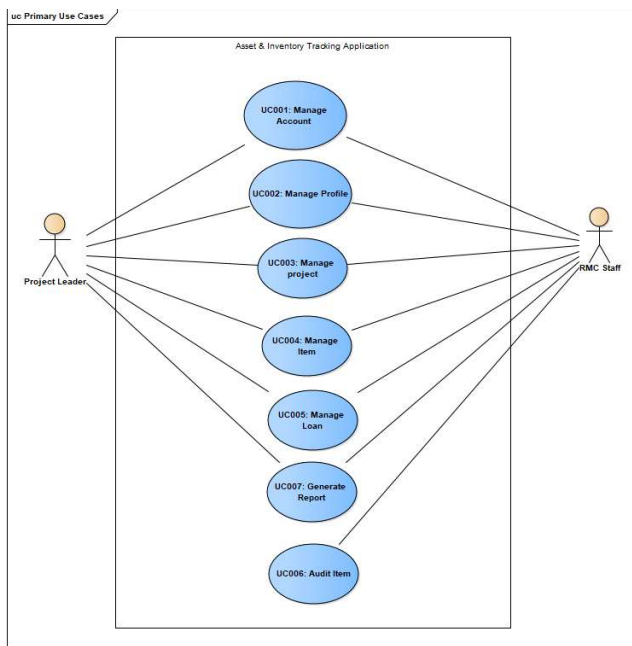


Figure 1. Use case diagram for IATA.

One of the key features is to manage team leaders' profiles in IATA as shown in Figure 2. This allows only registered users to use the application for asset and inventory management at RMC, UTM.

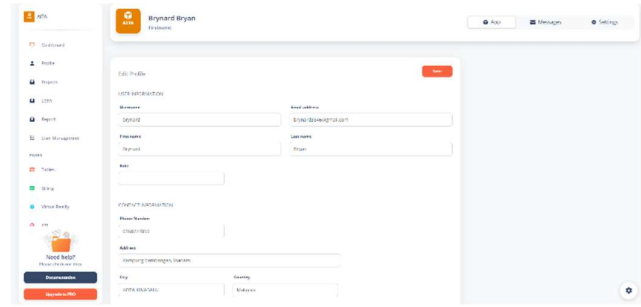


Figure 2. Profile page in IATA.

Figure 3 shows the add item page in IATA following the format used at RMC, UTM. This enables project leaders to register the assets or inventories under their projects.

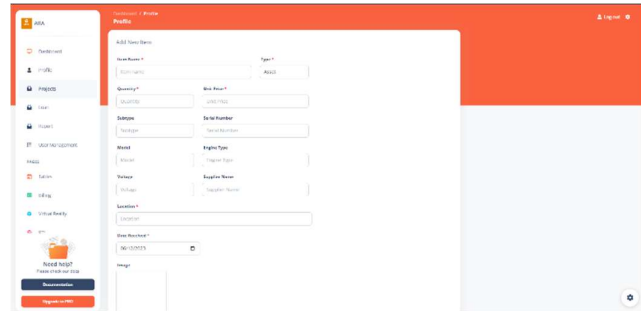


Figure 3. Add item page in IATA.

V. IMPLEMENTATION AND TESTING

System implementation involves developing a system or application based on the information gathered during requirement gathering, system analysis, and design phases. To ensure the system meets the desired requirements and is free of errors, system testing is performed. This test includes executing system components using manual or automated tools to identify any errors. In Laravel, the routes/web.php file serves as the entry point for handling Web requests. When a Web request is made, Laravel's routing system directs it to the appropriate route defined in this file. Laravel applies MVC architectural pattern that organizes the codes in respective modules or components to ease maintenance. Figure 4 shows the snippet of codes for controller in IATA.

System testing aims to validate the application against functional and non-functional requirements, minimizing errors. It includes black box testing, white box testing, and user acceptance testing. The subsequent section explores the testing methods utilized for the AITA. Black box testing is an approach where the tester focuses solely on input and output functionality without knowledge of the system's internal structure or implementation details. White box testing examines the internal structure, code, and logic of the system, leveraging the tester's knowledge of its internal workings to design and execute tests. All the tests conducted resulted in a pass and yielded positive outcomes.

```

class ProjectController extends Controller
{
  // public function index()
  // {
  //   $projects = Project::all();
  //   // dd('hello');
  //   return view('pages.projects', compact('projects'));
  // }

  public function index(Request $request)
  {
    $search = $request->input('search');
    $userId = auth()->user()->id; // Get the ID of the currently authenticated user

    $query = Project::where('user_id', $userId); // Filter projects by user ID

    if ($search) {
      $query->where(function ($q) use ($search) {
        $q->where('name', 'LIKE', '%' . $search . '%')
          ->orWhere('budget', 'LIKE', '%' . $search . '%')
          ->orWhere('status', 'LIKE', '%' . $search . '%');
      });
    }

    $projects = $query->orderBy('created_at', 'desc')->paginate(10);

    return view('pages.projects', ['projects' => $projects]);
  }
}

```

Figure 4. Snippet of codes for controller.

VI. CONCLUSION

The project objectives outlined have been achieved. This includes eliciting requirements for the inventory tracking applications through stakeholder meetings and documenting the Software Requirements Specification (SRS). AITA was designed based on these requirements, with the System Design Document (SDD) and user interface prototype created. The development phase successfully implemented AITA with proper functionality. Testing was conducted using suitable techniques, documented in the System Test Document (STD).

For future iterations of AITA, there are several enhancements that can be implemented to improve functionality and user experience. Firstly, multi-user support can be added to accommodate different roles, facilitating collaboration and access control. QR codes can be integrated for item tracking, allowing users to easily locate items within the inventory and provide proof of return. Additionally, enabling the generation of reports in Excel format would enhance reporting capabilities, enabling data analysis and seamless sharing of information with stakeholders.

These suggested enhancements aim to elevate IATA by incorporating multi-user support, QR code-based item tracking, and Excel report generation. By considering these improvements, AITA can offer a more comprehensive and efficient solution for managing assets and inventories.

REFERENCES

- [1] Vinculum, "Top 5 Reasons why you need Inventory Management". <https://www.vinculumgroup.com/the-ultimate-5-reasons-why-you-need-inventory-management/>, 2024.
- [2] A. Jenkins, "What is Inventory Management? Benefits, Types, & Techniques." <https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management.shtml>, 2024.
- [3] Sortly, "Simple Inventory Management Software for Small Businesses", <https://www.sortly.com/>, 2024.
- [4] Gocodes, "Inventory Tracking Software", <https://gocodes.com/>, 2024.
- [5] Stock and Inventory Simple. <https://play.google.com/store/apps/details?id=com.stockmanagement.next.app&hl=en&gl=US>, 2024.
- [6] Research Management Centre (RMC), "The Research Arm of Universiti Teknologi Malaysia (UTM)", <https://rmc.utm.my/>, 2024.
- [7] A. Singh, P. Chawla, and K. Singh. "Formulating an MVC Framework for Web Development in JAVA." *IEEE International Conference*, <https://ieeexplore.ieee.org/abstract/document/8553746>, 2018.

- [8] Lucidchart, <https://lucidchart.com/>, 2024.
- [9] Spring Boot, "Difference Between Spring MVC and Spring Boot." <https://www.interviewbit.com/blog/difference-between-spring-mvc-and-spring-boot/>, 2024.
- [10] VS Code, "Visual Studio Code - Code Editing. Redefined", <https://code.visualstudio.com/>, 2024.
- [11] MySQL, <https://www.mysql.com/>, 2024.