

Design and Development of CMMS Mobile Application for Manufacturing Factory

Nurshazreena Said Ramadhan¹, Shahreen Kasim^{1*}

¹Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2022.03.02.018>

Received 14 June 2022; Accepted 26 September 2022; Available online 30 November 2022

Abstract: Mobile CMMS for manufacturing factory is a mobile application to keep maintenance data for centralized access, automates tasks for increased efficiency, and aids in the management of maintenance procedures. With the development of the application, work order management and maintenance management will be more systematic and efficient. The development of this system uses Agile methodology and Scrum framework which includes 7 phases which is plan, iterate, analyze, design, develop, test and release phase. The development tools that will be used are Flutter, Android Studio and the database used is Firebase. The developed contains a login module, work order module, asset module and alert notification module. With the application, the employee is not tied to their desk, they can easily to update or make notes of the work order and maintenance while they're on site to keep the data more accurate.

Keywords: CMMS, Mobile application, Other Keywords

1. Introduction

A CMMS is an essential component of any maintenance management software toolkit, allowing users to manage work orders, inventory, maintenance operations, and more from a single, real-time platform. Any firm that has to maintain equipment, assets, or property can benefit from CMMS software. Some CMMS products are tailored to specific industries (e.g., the maintenance of factory machines, power plant or health care facilities). Other products are aim to be more general. A CMMS is a data-driven management solution, and the work order feeds that data. When an order is closed, the data is processed by the system. Work orders contain key asset maintenance data such as parts and costs, wrench time, asset repaired, and time to complete. This data is then processed into reports that are used by maintenance managers and other department heads to make data-driven decisions. The introduction of Computerized Maintenance Management Systems (CMMS) in the maintenance and facilities management areas has resulted in a significant change in the maintenance industry. This technology has assisted many businesses, ranging from small to large, in efficiently controlling and organizing their assets, and thus increasing asset uptime and longevity and lowering equipment maintenance costs.

*Corresponding author: shahreen@uthm.edu.my

The objectives of CMMS are to help organizations to track Preventive Maintenance schedule. Preventive maintenance implies to performing tasks or maintenance early, and it is possible to perform maintenance far too soon, resulting in higher costs and delays in actual needs. Other than that, the organization needs to balance maintenance personal workload. An unbalance maintenance personnel workload will cause stress and fatigue to the technician and thus can lead to injuries. Prioritization of tasks is a critical component of maintenance scheduling, which allows teams to focus on more challenging tasks. Lastly, many organizations struggling to identify an effective way to improve maintenance job. Maintenance management is essential for all businesses to effectively manage their assets and ensure smooth operations. Poor maintenance management can cause organizations to spend more money than necessary to keep their equipment operational.

2. Literature Review

2.1 Comparison with the Existing Systems

Table 1 shows the comparison of Evantage, Mpulse, iMaint and Mobile CMMS(Proposed system). Every application has work order management, preventive maintenance management and notification features. Other application does not have the predictive maintenance in their features compared with proposed application. The Mpulse CMMS, iMaint mobile and Mobile CMMS (proposed system) can adapt to any devices such as desktop and smartphone meanwhile Evantage CMMS only can use on web app and windows only. As for notification, it used to notify the maintenance team about the work order. Evantage and Mpulse CMMS notified maintenance team through the email while iMaint mobile and Mobile CMMS (proposed system) notified the team through their smartphone.

Mobile CMMS (Proposed system) are using predictive maintenance strategies to reduce cost and increasing asset's life meanwhile Mpulse, Evantage and iMaint are using preventive maintenance. Predictive maintenance (PdM) discovers, analyze and fix failures before it happens by comparing the trend of measured physical parameters against known engineering limits. These limits must be created in order for an issue to be discovered during routine monitoring. Correcting of the root problem is the key to most predictive efforts [1].

Table 1: System's Comparison

Features/System	Evantage	Mpulse	iMaint mobile	Mobile CMMS (proposed system)
Log in	√	√	√	√
Work order management	√	√	√	√
Preventive maintenance	√	√	√	X
Predictive maintenance	X	X	X	√
Calendar view	X	√	X	√
Notification	√ through email	√ through email	√ through smartphone	√ through smartphone
Adapt to any device	X Web app and windows only	√ Desktop and mobile	√	√ Desktop and mobile

3. Methodology/Framework

The project system development methodology that will be used is Agile development methodology. The Agile methodology breaking up a project management into several phases. It requires ongoing engagement with stakeholders as well as continuous improvement at all phases. Teams go through a planning, execution, and evaluation phase once the job begins. Both with team members and project stakeholders, continuous cooperation is essential. Agile is an iterative and incremental method, every defined phase's gets executed within a series of small iterative cycles of the software development process known as sprint [2]. Agile Development perfectly fits Mobile Software Development. There are plenty of ways confirming that agile techniques suites validation and experimentation of mobile applications [3]. According to new research, the benefit of agile methodology is improved software quality. Using Spike, the agile method conducts research to identify solutions to unknown problems; it works in fewer iterations, making it easier to test and validate [4]

3.1 Scrum Methodology Framework

Scrum is an agile methodology framework that allows to manage and control requirements along with software development. It is an iterative and incremental base model in which a software module is produced in many fragments in an iteration process. Scrum was created to improve the reproducibility of development processes, match individual and organizational dictums, build a culture based on repetition, aid shareholder value creation, have a positive message of performance at all levels, and improve distinct development and class of life. Scrum is a versatile approach that can be applied to any project in any sector. It may be used for both small and large projects [5].

3.2 Plan Phase

The planning phase's goal is to plan the project process and activities necessary to assure project success. A meeting between scrum masters and the stakeholders are conducted to acquire the objectives and requirements of the project. First step is to determine the project objective with the stakeholder. The project objective is to develop a CMMS mobile application for manufacturing factory. Next step was to scope out the problem and determine the solutions with the teams and stakeholder as well as identify cost and estimate time to complete the project.

After the objectives, requirement and estimated time to complete the project was acquired, the applicable technology and software and hardware requirement that will be used for the project was decided. The applicable technology that can be use in the project is mobile cross-platform technology. The cross-platform technology allowing the application to run in multiple mobile operating system such as Android and IOS using a single programming language.

3.3 Iterate

Scrum delivered using an incremental approach by working in iteration which known as sprint. The sprint begins with Sprint Planning where all team gathers and hold a meeting to determine which items from product backlog that can be achieve in the next sprint [6]. A meeting was hold through online using Google Meet as a medium.

3.3.1 Analyze

Before developing a product, it is critical to have a solid grasp of the product. In this phase, analysis of end user needs is made to ensure the application can meet their expectations. Any uncertainties must be clarified immediately during this step. After gathering requirements, an analysis is performed to determine whether the creation of a product is feasible. In the event of any uncertainty, a meeting is scheduled for additional discussion.

System requirements analysis is the process of determining the requirements that need to be met by a developed system, or the results of user expectations from a proposed system. It also focuses on activities that determine the requirements or conditions to meet the system by taking into account, analysing, documenting, verifying and managing system requirements. System requirements are divided into functional and non-functional requirements, user requirements and system requirements. Table 2 summarizes the functional modules provided in the system.

Table 2: System function module

No.	Module	Details	Users
1	Log in	Log in process into system	Technician
2	Work order	User update work order.	Technician
3	QR code	User can access to asset's details and update work order by scanning the QR code	Technician
4	Alert Notification	After work order is created, the application will create a notification to notify work order to the user's mobile phone.	Technician

4.1 Functional and non-functional requirement analysis

Functional requirements are features or functionalities that must be implemented by developers in order for users to complete their objectives. Table 3 shows the functional requirement for the system. Non-functional requirements are defined as global requirements such as system development or operating costs, performance, reliability, maintainability, portability and robustness. Table 4 shows non-functional requirement for the system [7].

Table 3: Functional Requirement

No	Module	Details
1	Log in	<ul style="list-style-type: none"> The system must allow the user to log into the system using the registered user ID and password. The system should only allow users with the correct user ID and password to enter the system. The system needs to be sensitive to users who enter the system using invalid user IDs and passwords. The system should take the user to the dashboard after a successful login.
2	Work Order	<ul style="list-style-type: none"> The system must allow technician to update their assigned work orders only.
3	Asset	<ul style="list-style-type: none"> The system must allow user to access asset in asset list page The system must allow user to access asset's details after scanning the QR code. The system must allow user to update work order after scanning QR code The list of work order must show asset's and assigned work order only.

Table 3

4	Alert Notification	<ul style="list-style-type: none"> The system must create notification to notify work order to the technician's mobile phone.
---	--------------------	--

Table 4 Non-functional requirement

No.	Requirements	Details
1	Compatibility	The application can be used in many operating system such as iOS and Android
2	Security	Security system allow access to accounts when users enter the correct username and password.
3	Usability	Users can easily can navigate the interface
4	Speed	The application is fast at responding to the command.

3.3.2 Design

The system is design in design phase to meet the requirements that were identified in previous phase. The requirement is converted into System design document that precisely represent the system architecture and can be utilized as an input to system development in the Development phase. The objective of design phase is to turn all of the requirement into detailed specification covering every area of the system and approval from client to progress to the Development phase.

In this phase, a business workflow and system flows were designed and presented to the stakeholder. Figure 1 shows system workflow of the proposed system. The workflows were evaluated and changes are made to ensure the workflows are accurate. After designing the workflow, the user interface (UI) was designed to accomplish and make sure the features are simple to access and easy to understand as well as it utilized to make tasks easier. Appendix A shows initial UI design for the application.

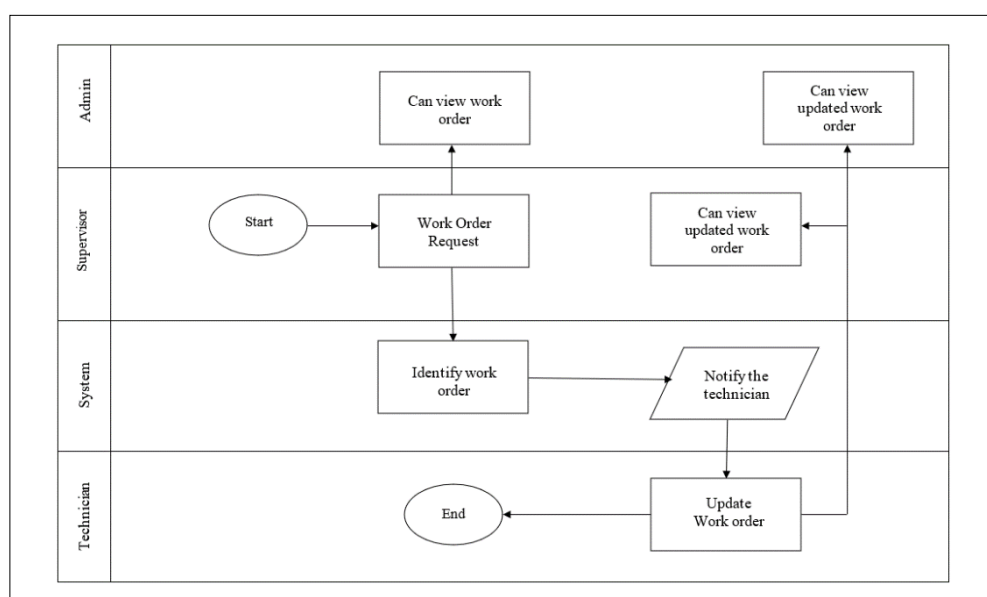
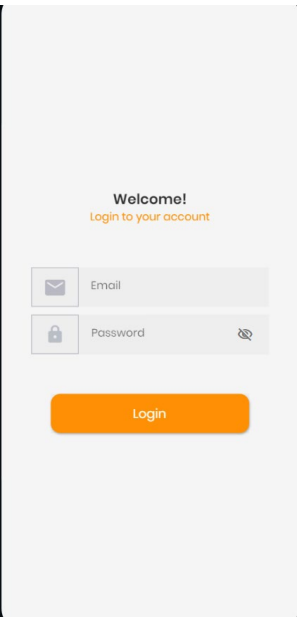


Figure 1 Proposed system workflow model

3.3.3 Develop

This is the part where developer starts to codes and build the application refers to the design phase. The programming language that will be used to develop the application is Dart language. Dart is a client-optimized programming language used to develop fast apps on any platform such as iOS and Android. In the development of a Computerized Maintenance Management System, the deployment and testing phase is crucial. Its goal is to make sure the system meets the goals set out in the prior planning phase. Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8 shows the application interface and a snippet of code for the interfaces,

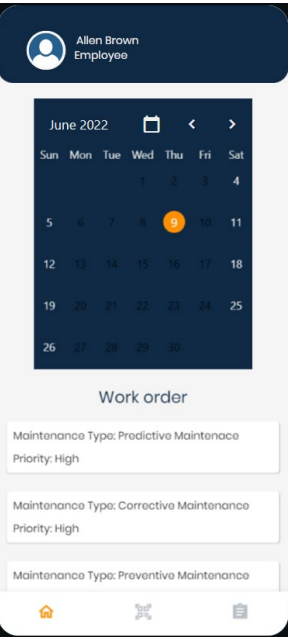


```

child: FFButtonWidget(
  onPressed: () async {
    final user = await signInWithEmail(
      context,
      emailController.text,
      passwordController.text,
    );
    if (user == null) {
      return;
    }
    await Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => DashboardWidget(
          userid: currentUserUid, // passing data to dashboard
        ),
      ),
    );
  },
)

```

Figure 2 Login Interface



```

Expanded(
  child: StreamBuilder<List<WorkOrderRecord>>(
    stream: queryWorkOrderRecord(
      queryBuilder: (workOrderRecord) => workOrderRecord // collection of database
        .where('assignedto', isEqualTo: currentUserUid) // filter
    ),
    List<WorkOrderRecord> listViewWorkOrderRecordList = snapshot.data;
    return ListView.builder(
      padding: EdgeInsets.zero,
      scrollDirection: Axis.vertical,
      itemCount: listViewWorkOrderRecordList.length,
      itemBuilder: (context, listViewIndex) {

        final listViewWorkOrderRecord =
          listViewWorkOrderRecordList[listViewIndex];

```

Figure 3 Dashboard Interface

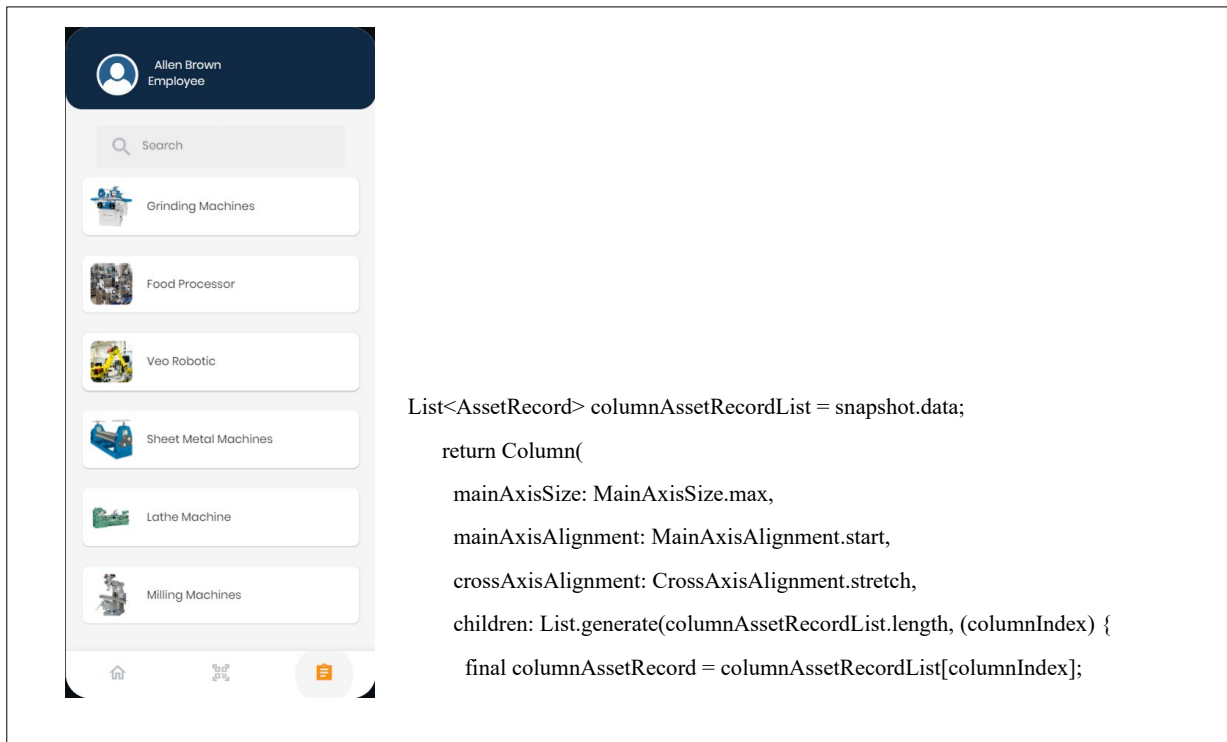


Figure 4 Asset list Interface

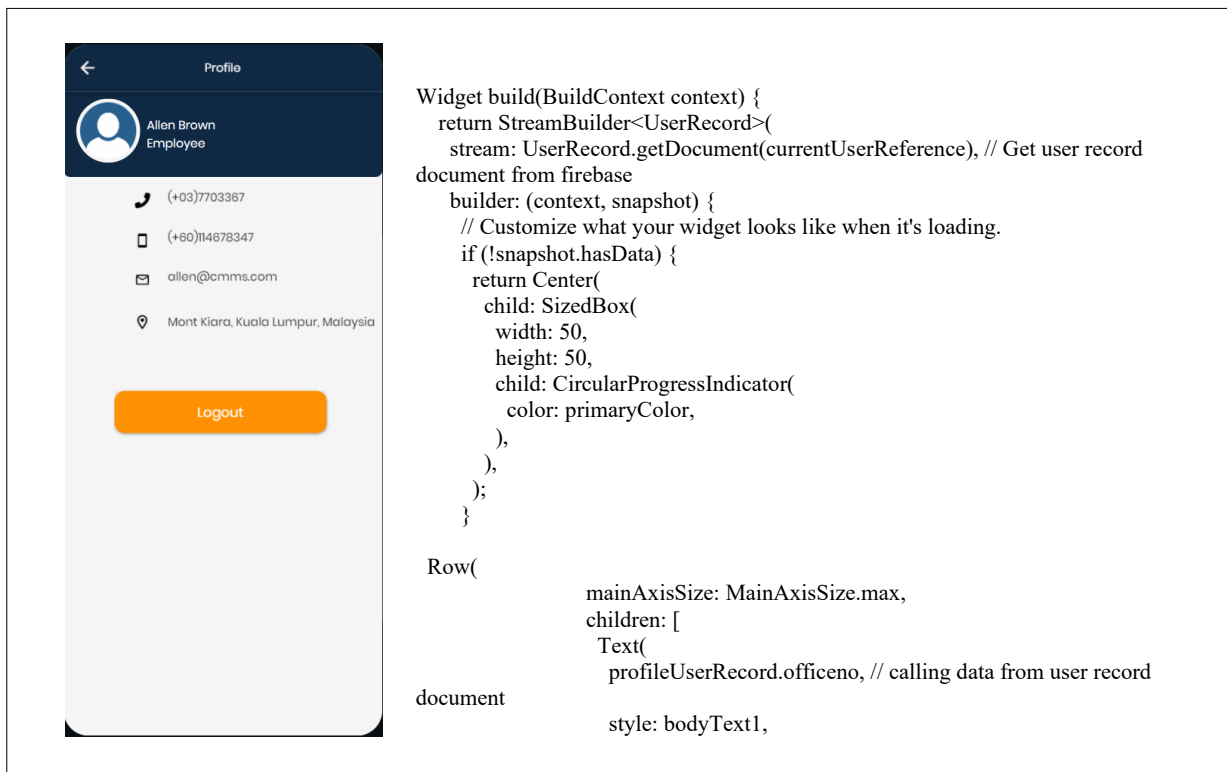
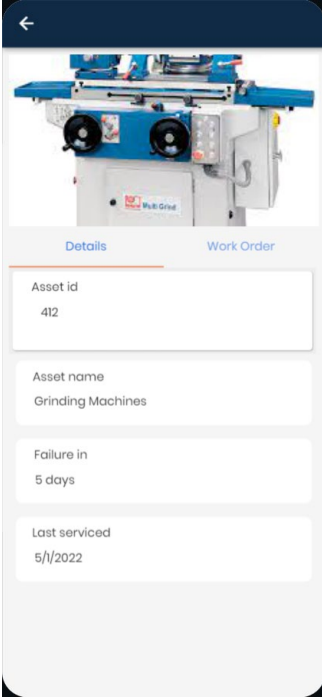


Figure 5 User Profile Interface

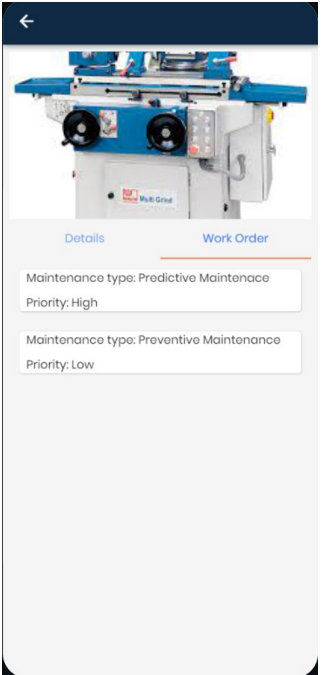


```

@Override
Widget build(BuildContext context) {
  return StreamBuilder<List<AssetRecord>>(
    stream: queryAssetRecord(
      queryBuilder: (assetRecord) => assetRecord
        .where('qrcode', isEqualTo: widget.qrcodeValue) / filter
        .where('asset_id', isEqualTo: widget.id), // filter
      singleRecord: true,
    ), // connect with database
  );
}

```

Figure 6 Asset Interface



```

StreamBuilder<List<WorkOrderRecord>>(
  stream: queryWorkOrderRecord(
    queryBuilder: (workOrderRecord) => // connect to database
      workOrderRecord
        .where('assignedto', isEqualTo: currentUserUid ) //
filter/condition
        .where('status', isNotEqualTo: 'Done')
        .where('asset_id', isEqualTo: widget.id),
  ),
)

```

Figure 7 Asset's Work Order Interface

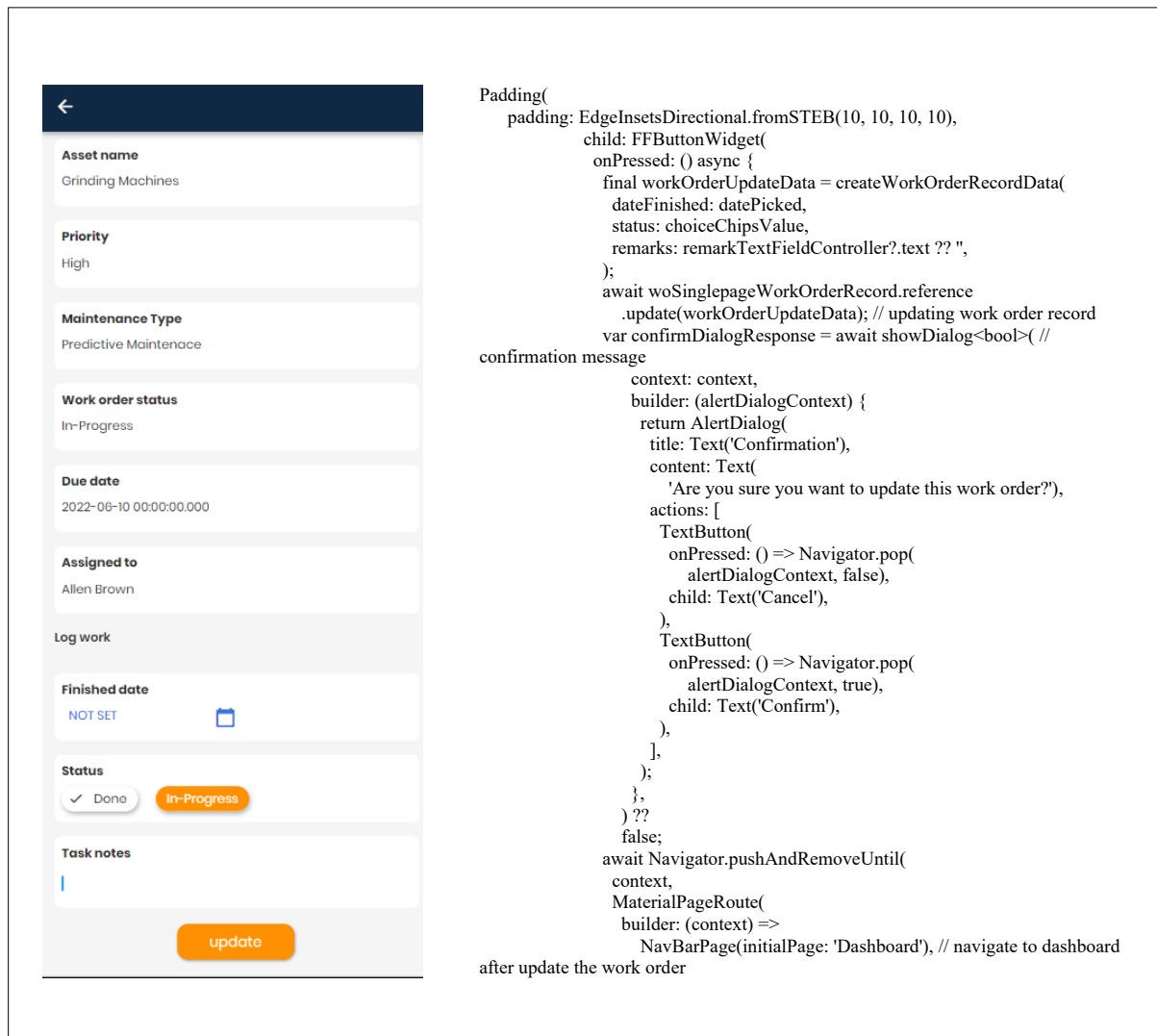


Figure 8 Work Order Interface

3.3.4 Test

After development phase, the application must be tested to ensure it met all the requirements. A test plan is created by the project team before testing can begins. During the testing phase, many types of testing are performed, including quality assurance testing (QA) and user acceptability testing (UAT). Other than that, functional and non-functional requirement of the system need to be tested. Table 5 contains a list of the system testing forms and the results of the tests of the system modules. Tests are a crucial part of developing a system. A system's functionality is tested to ensure that it runs smoothly. Any errors or mistakes will be corrected. In order to ensure that the system developed meets the needs of the users, system testing should be evaluated. All module functions were tested and found to function according to set objectives. During the testing, some errors were found that needed to be fixed. Once the error was corrected, the final results showed that the system had achieved its objectives.

Table 5 List of test cases

Item	Test cases	Expected Outcome	Actual Outcome
1	Login Module		
	Login to the system using the email and password that have been registered into the database.	Able to login into the system using registered email and password.	Success
	Error message displayed when user enter wrong email format.	“Email was badly formatted” error message is displayed.	Success
2	Work Order Module		
	The dashboard listed out all work order that were assigned to the user	A list of work order that was assigned to the user were listed in the dashboard.	Success
	The list of work order for each asset	A list of work order was listed for every assets which are assigned to the user.	Success
	Work order details	The page shows all the work order details.	Success
	User can update the work order	The work order was updated by fill the finished date, status and notes.	Success
	A confirmation message was displayed when user update the work order	A confirmation message “Are you sure you want to update the work order?” was displayed.	Success
3	Asset Module		
	List of Asset	User can view asset list in asset list page.	Success
	Search asset	User can search asset using the search function.	Success
	Scan QR code	User can scan QR code to access the assets details and work order .	Success
4	Alert Notification		
	Alert notification	User gets notification after a work order has been added.	Success

4. Result and discussion

The developed system has been tested for user acceptance. This survey aims to obtain comments and suggestions to assist in improving the application. Therefore, an evaluation form was provided to the users during this testing. It can be concluded from the evaluation form that the user is satisfied with the navigation layout, menu icons, and text formatting.

The feedback collected has shown that the users are satisfied with the proposed system. Figure 9 shows the level of user satisfaction and acceptance of the system developed based on the feedback collected. The testing process, whether alpha or beta, can help system developers by collecting user

feedback. As a result of this testing, some suggestions and improvements suggested by users will likely improve the system's quality in the future.

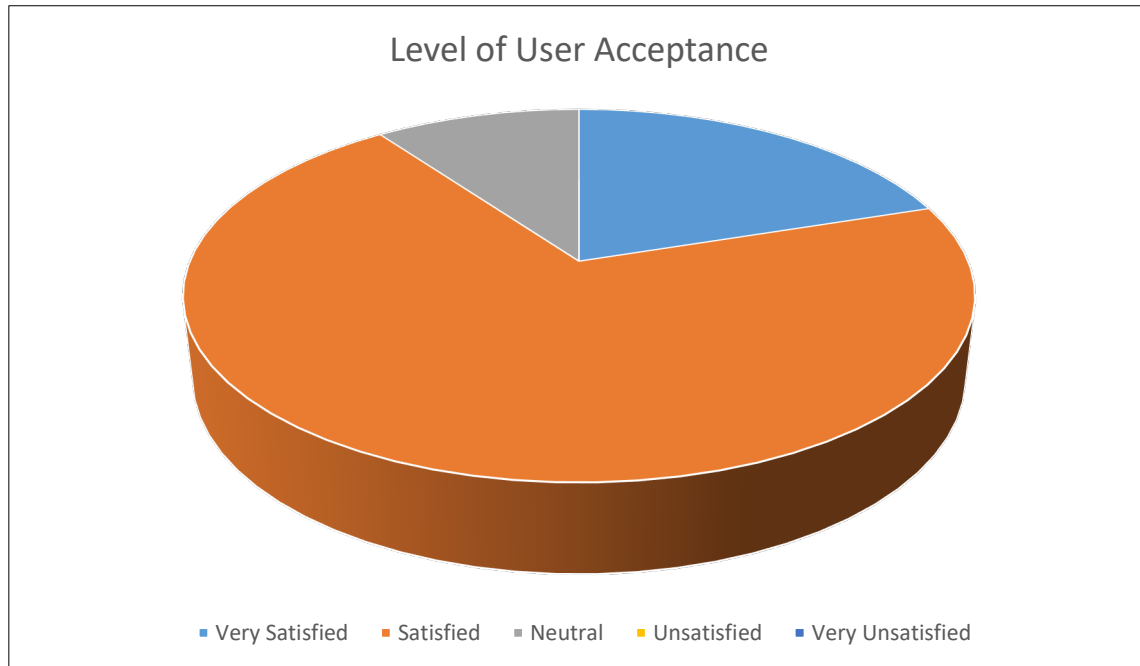


Figure 9 Level of User Acceptance

5. Conclusion

To conclude, it is clear that the system has successfully been developed and has met the objectives and scope defined at the beginning of the project planning process. By using Agile methodology and Scrum framework, the development of the application can be fulfilled before the due date. This project has met its objectives and provided an overview of its strengths and limitations. As several suggestions were made for future improvements to the Mobile CMMS application, the application is expected to continue to improve in the future. The following are several suggestions that can be implement to improve the application in the future:

- i. The application can include important variables such as temperature, health, volume and many more in asset's detail.
- ii. The application can organize work order in priority and due date.
- iii. The application can show how many overdue work orders and list them out.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix

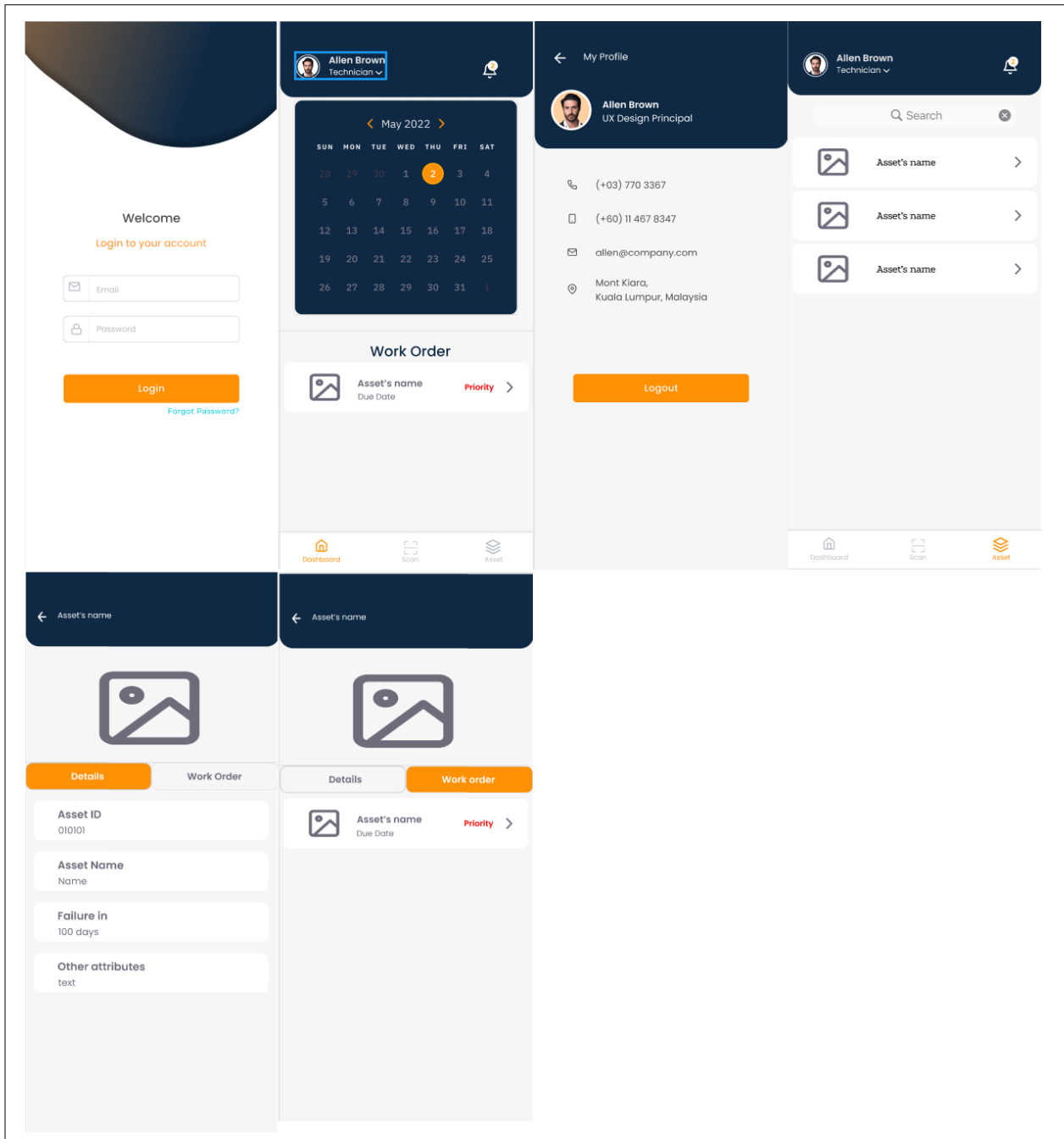


Figure 10 UI design

References

- [1] A. Jezzini, M. Ayache, L. Elkhansa, B. Makki, and M. Zein, "Effects of predictive maintenance(PdM), Proactive maintenace(PoM) & Preventive maintenance(PM) on minimizing the faults in medical instruments," in *2013 2nd International Conference on Advances in Biomedical Engineering*, 2013.
- [2] P. Jain, A. Sharma, and L. Ahuja, "The impact of agile software development process on the quality of software product," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2018.
- [3] B. Mathur and S. M. Satapathy, "An analytical comparison of mobile application development using agile methodologies," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019.
- [4] A. R. Chaudhari and S. D. Joshi, "Study of effect of agile software development methodology on software development process," *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2021.
- [5] F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab, and M. Abbas, "The influence of agile methodology (scrum) on software project management," in *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2019.
- [6] P. Flewelling, *The Agile Developer's Handbook: Get more value from your software development: get the best out of the Agile methodology*. Birmingham, England: Packt Publishing, 2018.
- [7] D. Ameller, X. Burgués, D. Costal, C. Farré, and X. Franch, "Non-functional requirements in model-driven development of service-oriented architectures," *Sci. Comput. Program.*, vol. 168, pp. 18–37, 2018.