

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module** | EG3611A (IA) | | + EG3611B (IA) | | | EG3612 (VIA) |
|  | EG1611 (Co-Op) | | EG2620 (Co-Op) | | |  |
|  |  | |  | | |  |
| **Report** | Interim | | Final | | |  |
|  |  | |  | | |  |
| **Reporting Period** | 17/5/2021 | to | | 13/8/2021 |  | |
|  |  | | | | | |
|  |  | | | | | |
|  |  | | | | | |
|  |  | | | | | |
| **Name** | **Goh Kheng Xi, Jevan** | | | | | |
|  |  | | | | | |
| **Matriculation No.** | **A0199806L** | | | | | |
|  |  | | | | | |
| **Department** | **Engineering Science** | | | | | |
|  |  | | | | | |
| **Internship Period** | **17/5/2021 to 13/8/2021** | | | | | |
|  |  | | | | | |
| **Mentor** | **Van Kan, Jeroen Anton** | | | | | |
|  |  | | | | | |
| **Company** | **Helios Applied Systems Pte Ltd** | | | | | |

Contents

[1. Acknowledgement 2](#_Toc79688959)

[2. Introduction 2](#_Toc79688960)

[2.1 about employer 2](#_Toc79688961)

[2.2 Objectives and motivations 3](#_Toc79688962)

[3. Tools monitoring project 3](#_Toc79688963)

[3.1 Project aim 3](#_Toc79688964)

[3.2 Qt 3](#_Toc79688965)

[3.3 Smartphone application 3](#_Toc79688966)

[3.3.1 Functions 4](#_Toc79688967)

[3.3.2 Pages and dialogs 4](#_Toc79688968)

[4. Block stacking project 7](#_Toc79688969)

[4.1 Project aim and purpose 8](#_Toc79688970)

[4.2 Object-oriented programming (OOP) 8](#_Toc79688971)

[4.3 Property bindings (Signals and Slots) 8](#_Toc79688972)

[5. Challenges and obstacles 9](#_Toc79688973)

[6. Reflections 9](#_Toc79688974)

[7. References 9](#_Toc79688975)

[8. Weekly journal 10](#_Toc79688976)

# 1. Acknowledgement

I would like to extend my appreciation and gratitude to Helios Applied Systems for accepting me into the internship programme and Dr Shyi Herng for his guidance and teachings throughout the programme, inspiring and exposing myself to the engineering and technological industry. I would also like to thank all my teammates for their assistance and feedback that enhanced the overall internship experience, cultivating my team communication skills and developing my technical competency.

# 2. Introduction

## 2.1 about employer

Helios Applied Systems Pte Ltd is a company that engineer technological solutions for customers from various industries such as healthcare, semiconductors, and advance manufacturing [1]. Solutions often involves the designing of specialised devices, instrumentation, artificial intelligence, and image processing [1].

## 2.2 Objectives and motivations

One of the internship objectives is to gain greater exposure in the technology and engineering industry. Industrial exposure includes understanding project organisation and project workflow in the working environment in addition to knowing the relevant industrial skills required.

Other objectives include honing of current technical skills and improving of programming languages proficiency, in addition to developing soft skills such as better time management, communication, and organisation skills.

# 3. Tools monitoring project

## 3.1 Project aim

The aim of the given project is to design and develop a software product for inventory monitoring in the aeronautical industry. The myriad of tools in the hangers, together with the hectic workflow and high traffic of tools in the workplace, it is difficult to keep track of them and hence, Helios applied system is designing a solution in the form of a software.

The product includes a smartphone application and a desktop application where the smartphone application will allow the ground engineers to take images of the toolboxes with their respective tools when they retrieve and return the tools from the toolboxes. The images taken are then sent to a computer for image processing. The image processing software will then identify the tools present and the tools missing and output a pdf report for the engineers to view.

My main contribution for this project is the smartphone application component.

## 3.2 Qt

Qt is an application development framework that provides a cross-platform software development toolkit [2]. Applications can be programmed via C++ programming language or QML (Qt Modelling Language). Another commonly used programming language in Qt is python through python bindings provided by Qt. However, in this project, only C++ and QML were used.

QML is a declarative language that allows for C++ extensions and supports JavaScript expressions.

Use of Qt is relatively easy and well-documented compared to many other cross-platform development software. Even though the setting up and configuring of Qt for android applications was a bit of a hassle and complex, the layout and navigations of the Qt Creator IDE (integrated development environment) are quick to adapt to.

## 3.3 Smartphone application

For the smartphone application, QML was used to programme the layout and the user interface of the application.

As QML has no support for the reading and writing of files, C++ extensions were used by creating a QObject class and registering it as a QML component.

Databases and application data were set up using JSON (JavaScript Object Notation) and the details in the user interface were populated using JavaScript expressions.

This following section describes a non-exhaustive list of the functions and features of the smartphone applications. A much more detailed documentation is written for the smartphone application towards the end of the internship.

### 3.3.1 Functions

The smartphone application serves 2 main purposes which are to take the image of the tools to be exported to the desktop server for image processing, and to view the database (employee and tools data).

### 3.3.2 Pages and dialogs

The smartphone application includes the following pages:

1. Login page (with barcode scanner)
2. Menu page
3. Camera page and image gallery
4. Database page (for admin users)
5. Employee database page

A picture containing text

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a cell phone

Description automatically generated with low confidence

Figure 3. Password dialog

Figure 2. Barcode scanner

Figure 1. Login page

The login page (Figure 1) consists of a text input for the user to input their employee pass ID and a start button to confirm the employee pass ID. If the employee ID does not exist in the database, a red “INVALID” text will be shown above the input box. A barcode scanner button is implemented below the input box for the employee to scan (Figure 2) their employee pass ID as a possible convenient alternative to manually typing in their pass ID. For both methods, if the employee ID exists in the database, a password dialog (Figure 3) will be displayed for the user to key in their 6-digit pin for security.

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

Figure 5. Menu page(admin)

Figure 4. Menu page(non-admin)

The menu page (Figure 4) is the first page that the user will interact with once the user is logged in. The actions allowed on this page depends on the admin status of the user. For both admins and non admins, there is a camera button to go to the camera page and a database button to view the database. However, for admins, there are additional import and export buttons (Figure 5) for them to import and export the respective database.

A screenshot of a cell phone

Description automatically generated with medium confidenceA close up of a person's skin

Description automatically generated with low confidence

Figure 7. Preview page after pressing capture button

Figure 6. Camera page

The camera page (Figure 6) consists of two scrollable rows at the top of the page that allows the user to select the toolbox and the drawer for the image to be taken. Focusing of the camera is done by tapping on the viewfinder and capturing of the image (Figure 7) is done by pressing on the capture button below the middle of the viewfinder. Additional features include a zoom function and the camera light switch. Pressing on the image album preview icon on the bottom right of the screen brings up the image gallery.

A screenshot of a cell phone

Description automatically generated with medium confidenceA picture containing text

Description automatically generated

Figure 9. Pressing once hides rows and displays delete button

Figure 8. Image gallery

Text

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedIn the image gallery (Figure 8), the images can be navigated by scrolling the same scrollable rows in the camera page to view the Images of the desired toolbox and drawer. Additional features include a delete button (Figure 9) and a row of the drawer images for easier navigations and multiple images viewing.

Figure 12. Employee Database page

The database page can be accessed through the database button in the menu page. However, the database page differs between admin users and non-admin user. For admin users, the database page displays the entire list of employees and their details (Figure 10). By pressing on a specific employee, collapsible delegates representing the toolboxes with their respective drawers and tools that the employee owns are displayed. Pressing on a tool twice brings up a dialog (Figure 11) with more information on the tool such as the quantity accounted for and the total quantity supposedly in the drawer. Pressing on an employee twice brings up the employee database (Figure 12) analogous to the page displayed if a non-admin user were to press the database button in the menu page. In this employee database page, all details of the employee are displayed. The purpose for the different database for admin and non-admin user is to only allow admins to view the entire database while non-admins can only view their own data.

Figure 11. Tool Dialog

Figure 10. Admin database page

# 4. Block stacking project

As the smartphone application was completed relatively fast, I went over to the block stacking project for about 2 weeks while waiting for the remaining components in the tools monitoring project to be completed. My main contributions for the project are the setting up of the coding infrastructure for Object-Oriented programming.

## 4.1 Project aim and purpose

The aim of this project is to design and program a block stacking algorithm in C# programming language. The algorithm takes in a list of blocks of different predefined dimensions and stacks them optimally in a volume of a given surface area and infinite height.

The algorithm is a part of a bigger project related to virtual reality environment, where boundary conditions and equations can be applied to the blocks in order to simulate the virtual environment like finite element analysis.

Some of the constraint requirements include:

1. Allows overhanging of not more than 50% of the block surface area
2. Fill each layer of the volume by alternating the corner the filling starts, top left corner and bottom right corner
3. Fill each column with alternating the orientation of filling, row by row and column by column

## 4.2 Object-oriented programming (OOP)

OOP is a powerful programming paradigm that allows the programmer to define new data structure and model objects by describing their attributes and interacting behaviour [3]. Benefits include better code development and maintainability due to it modular structure, allowing simultaneous changes to the code [3]. However, OOP often requires more memory and more computational operations (hence, higher compute time) than the other programming counterparts due to its modular nature [3].

For this algorithm, 4 classes, which are blueprints describing the objects, are created. The first is the “Block” class with attributes name, breadth, width and height. The second class is the “Cell” class which simulates each individual point in the 3D volume. The third class is the “Layer” class which consists of a 2-dimenional array of Cell objects, representing each horizontal plane at every height of the 3D volume. Lastly, the “Space” class consists of a list of Layer objects to model the 3D volume to be filled.

## 4.3 Property bindings (Signals and Slots)

Signals and slots is a mechanism, similar to the event listener concept from Java and JavaScript, used in Qt where components communicate with one another. For example, a change in the value of a component will trigger a behaviour of another component with the corresponding slot that is tied to that value change signal. Property bindings is a signals and slots mechanism used in QML where the value of a component’s property is “bound” to the value of another property from either the same or different component.

Bringing over this concept to the OOP algorithm, the property values of some classes are bound to another. For example, the “name” property of the “Block” class is bound to the “occupied” property of the “Cell” class. In addition, the “x”, “y”, “z” property of the “Block” class, will trigger a change in the “block” property of the “Cell” class.

# 5. Challenges and obstacles

Despite the scale of both projects being relatively small, they still pose considerable challenges due to the time constraint and the first-timer experience in the Qt toolkit. In the light of the ongoing pandemic during the internship, communication and interactions with the other team members, our employer and the product’s customer are limited. The inconvenience of online meeting severely undermined the discussion for the algorithms and codes in addition to the format and layout of the program and database. This causes discrepancy in the codes among the members with each members holding different versions of the code.

To overcome these adversaries, the team met up almost every morning to update the individual progress and to align the roadmap for the program. The separation of the tools inventory keeping software into different components allowed for parallel development of the software with minimal disparity and contradiction.

# 6. Reflections

The tools inventory keeping software have exposed the art of Graphic User Interface (GUI) and user experience (UX) programming which is underappreciated. In our day-to-day life in the virtual world, the seemingly simple and intuitive design interface that we did not give much thought to, sits atop sets of complex algorithms and codes. After the internship, I realised that the more intuitive an interface is, the more complex and effort the codes behind the interface are.

GUI programming is instrumental to engineers as it is the direct point of access between the application and the user. It defines the experience of the user and hence, the quality and expectation of the product.

In addition to GUI/UX programming, the internship has trained me to be a more adaptable and proficient programmer. Being in a team of engineers, each with different programming methodology, it is of paramount importance that we are adaptable with different methodologies and conventions.

This internship has also taught me about time management. As the internship lasted only 3 months, it Is practically unrealistic to push out the full complete product, and hence, there is a need for prioritisation and determining which component and functionalities should be completed first.

# 7. References

|  |  |
| --- | --- |
| [1] | Helios Applied Systems Pte Ltd, WordPress, [Online]. Available: https://www.heliosappliedsys.com/. [Accessed 11 8 2021]. |
| [2] | Qt, “Qt,” [Online]. Available: https://www.qt.io/. [Accessed 11 8 2021]. |
| [3] | B. Miller, “Green Garage,” 1 11 2017. [Online]. Available: https://greengarageblog.org/6-pros-and-cons-of-object-oriented-programming. [Accessed 11 8 2021]. |

# 8. Weekly journal

Week 1

* Installation of Qt
* Reading and watching of Qt tutorial
* Practicing of Qt C++
* Introduction to function block:  a concept of breaking down a project into different function blocks with their own respective input and output to understand a complex better
* Installation of android studio, emulators, and the related SDKs

Week 2

* Implemented a login page
* Finding a barcode scanner filter online
* Implementing barcode scanner into the login page which causes the switching of the codes from Qt C++ style to Qt QML
* Designing of database
* Populating of database
* Displaying of database in a scrollable list view

Week 3

* Designing of menu page and admin database page layout
* Changes in database -> refactoring codes
* Debugging of UI
* Downloading of JSONcons to convert JSON to CSV and CSV to JSON
* Implementing the import and export of database
* Testing and debugging of import and export of database
* Aligning algorithm and logic with pc server

Week 4

* Changing of login page UI and logic flow
* Adding of timer for invalid text
* Designing of camera page
* Coding of the scrollable row in camera page
* Designing of dialog for import and export function
* Implementing of capture button and zoom function

Week 5

* Changing of database and refactoring of codes
* Making collapsible sections in database page
* Debugging of camera (does not turn on when page is exited and re-entered)
* Changing of UI for dialog
* Changing of colors for database sections (toolbox rectangles and drawer rectangles when collapsed and expanded)
* Debugging of login page (Employee ID stays on login page when “logged out”)

Week 6

* Creating database for images and their metadata
* Implementing image and metadata storing algorithm in capture button of camera page
* Creating the image gallery and displaying the images in the image gallery
* Creating the deletion algorithm for the delete button in image gallery
* Photoshopping the delete button icon
* Implementing dialog for tools
* Bug fixing for changing of page when app is minimised
* Integration of smartphone application with pc server

Week 7

* Creating the delete button for the image gallery
* Implementing the camera flash button
* Implementing search function for database
* Changing of UI for scrollable rows
* Adding effects to dialogs (adding blur and darken to make dialogs more protruding)
* Helped in block stacking algorithm for the KKH project
* Installation of C#
* Coded the backbone for OOP in the KKH project
* Coded the four classes: Block, Cell, Layer, Space
* Coded the required functions (addBlock(), checkBlockAddable(), fillLayer(), etc..)

Week 8

* Visiting of engineers at Changi Airport
* Implementing 6-digit pin
* Changing of database format and refactoring codes
* Changing of UI for database page
* Debugging of block stacking algorithm (Layers does not get filled)
* Debugging of block stacking algorithm (Runs infinite loop when certain conditions are met)
* Dynamic components of database page for better user experience
* Aligning database with pc server

Week 9

* Changing of barcode scanner page layout
* Changing of UI for importing and exporting dialog (highlights, colours and text)
* Adding import success and failure feedback for importing and exporting
* Design of preview page after image is captured
* Designing of buttons using photoshop for the preview page
* Coding of logic after image is captured
* Scrollbar for database page
* Adjusting the output for the block stacking algorithm
* Debugging of functions for block stacking algorithm

Week 10

* Changing of UI for database page (sizing, text formatting)
* Readjusting of colours (more readable text)
* Repositioning of UI components
* Smoothening of UI
* Smoothen swiping gesture for scrollable rows
* Smoothen swiping gesture for images in image gallery
* Aligning codes with pc server

Week 11

* Doing of product demonstration video
* Editing and captioning of video
* Formatting video into different resolutions and size
* Writing of documentation for product
* Annotating of codes for both tool inventory keeping software project and KKH project
* Adding “status” key to images metadata
* Implementing logging feature for images

Week 12

* Editing of documentation and video
* Final changes to code for tools inventory keeping software project
* Bug fixing for database page (quantity of tools)
* Passing on of information and products to company’s employee
* Discussing of products future features
* Retaking of video and re-editing of demo video

Week 13

* Final debugging for overall codes
* Break and prepare for school reopening
* Passing on of information and products to company’s employee