

Florida International University
Knight Foundation School of Computing and Information Sciences

Capstone II

Final Deliverable

FIU IoTConnect

Fall 2023

Team Members: Zipporah Thomas, Khaled Khashashneh Salazar, Michael Simon

Product Owner(s): Selcuk Uluagac , Esteban Luques

Professor: Masoud Sadjadi

Abstract

This document presents the information necessary to gain understanding of our prototype application along with the Inventory Management System

This is a Mobile Application - Streamlined Check-In and Check-Out:

The FIU IoTConnect Mobile Application. Designed with the dynamic academic environment in mind, this app redefines the way students interact with the Internet of Things.

Table of Contents

Introduction

- What is FIU IoTConnect?..... 4

User Stories

- Implemented/Unimplemented User Stories..... 5-6

Project Plan

Technologies Used..... 7-8

Overview of Sprints Plan..... 9-10

System Design

Mobile Application 11

 Login Page. 11

 IoT Device Interface 12

Inventory Management System13 -14

Appendix

Appendix A: Database Entity Relation Diagram 15

Appendix B: Sprint Review Meeting..... 16 - 18

Appendix C: Maintenance, Shortcomings/Wishlist 19

References 20

What is FIU IoTConnect?

Purpose:

The FIU IoTConnect mobile application is designed to facilitate the management of Internet of Things (IoT) devices for students at Florida International University. It serves as a centralized platform for students to check in and check out IoT devices, promoting efficient utilization and monitoring of these resources.

Key Features:

1. Device Inventory:

- View a comprehensive list of available IoT devices.
- Check the availability status of each device.

2. Check In/Check Out:

- Reserve and check out IoT devices for specific durations.
- Check in devices when they are no longer needed.

4. Device Details:

- Access detailed information about each IoT device, including specifications and usage guidelines.

5. User Authentication:

- Secure login using FIU student credentials to ensure authorized access.

Database Architecture:

1. Device Database:

- Stores information about each IoT device, including type, specifications, availability status, and usage history.

2. User Database:

- Manages student profiles, authentication details, and reservation history.

3. Transactional Database:

- Records transactions related to device check-ins and check-outs.

Security Measures:

1. Authentication:

- Secure authentication using FIU student credentials, ensuring only authorized users can access the application.

2. Encryption:

- All data transmission between the mobile application and the server is encrypted to protect sensitive information.

User Stories

Implemented/Unimplemented User Stories

While reading some User Stories have been Implemented and some should be implemented for the next part of this project

User - Student:

1. **User Registration: Implemented**

- Students should be able to create an account using my FIU credentials so that I can access and manage IoT devices.

2. **View Device Inventory: Implemented**

- Students should be able to the available IoT devices to see what is currently in stock.

3. **Check Out Device: Implemented**

- Students should be able to reserve and check out an IoT device for a specific duration for my academic projects.

4. **Check In Device: Implemented**

- Students should be able to return the IoT device I borrowed when I'm done using it.

5. **Receive Notifications: Unimplemented**

- Students should be able to receive notifications reminding me to return the IoT device before the check-in deadline.

6. **View Reservation History: Unimplemented**

- Students should be able to view my past reservations and check-in/check-out history for reference.

User - Admin

1. **Manage Device Inventory: Implemented**

- For Admin user they should be able to add new IoT devices to the inventory and update existing device information.

2. **Monitor Device Availability: Implemented**

- For Admin user they should be able to see real-time information on device availability and usage.

3. **Generate Reports: Unimplemented**

- For Admin user they should be able to generate reports on device usage, popular devices, and any overdue check-ins.

4. **User Management: Unimplemented**

- For Admin user they should be able to manage user accounts, including the ability to add or deactivate accounts.

5. **Security Configuration: Implemented**

- For Admin user they should be able to configure security settings, including role-based access control and authentication methods.

6. **Handle Device Issues: Unimplemented**

- For Admin user they should be able to address any reported issues with IoT devices, such as malfunctions or damage.

User - System Administrator:

1. **Database Maintenance: Unimplemented**

- As a system administrator, I want to perform routine maintenance on the IoT device database to ensure data integrity.

2. **Backup and Restore: Unimplemented**

- As a system administrator, I want to regularly back up the database and have the ability to restore it in case of data loss.

3. **Security Audits: Unimplemented**

- As a system administrator, I want to conduct regular security audits to identify and address potential vulnerabilities.

4. **Software Updates: Unimplemented**

- As a system administrator, I want to ensure that the software and security protocols are up to date.

Technologies used

1. **Flutter:**

- Flutter is an open-source UI software development toolkit created by Google. It is used to build natively compiled applications for mobile, web, and desktop from a single codebase. Flutter uses the Dart programming language.

2. **Android Studio:**

- Android Studio is the official integrated development environment (IDE) for Android app development. It is based on JetBrains' IntelliJ IDEA and is used by developers to build Android applications in Java, Kotlin, and C++.

3. **Flask:**

- Flask is a lightweight and web framework for Python. It is designed to be simple and easy to use, making it an excellent choice for building web applications and APIs. Flask follows the WSGI (Web Server Gateway Interface) standard.

4. **JSON Web Token (JWT):**

- JSON Web Token is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. It is commonly used for authentication and information exchange in web development.

5. **Python:**

- Python is a high-level, interpreted programming language known for its readability and versatility. It supports multiple programming paradigms and is widely used in web development, data science, artificial intelligence, and more.

6. **CSS (Cascading Style Sheets):**

- CSS is a style sheet language used for describing the look and formatting of a document written in HTML or XML. It controls the visual presentation of web pages, including layout, colors, and fonts.

7. **HTML (Hypertext Markup Language):**

- HTML is the standard markup language for creating web pages and web applications. It structures content on the web and is essential for building the basic framework of a webpage.

8. **Java:**

- Java is a versatile, object-oriented programming language commonly used for building large-scale enterprise applications, mobile applications (Android), and web applications. It is known for its portability and platform independence.

9. **JavaScript:**

- JavaScript is a scripting language that enables the creation of dynamic and interactive content on web pages. It is commonly used for client-side development and is supported by most web browsers.

10. MSSQL (Microsoft SQL Server):

- Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is used for storing and retrieving data as requested by other software applications.

11. API Keys:

- API keys are codes passed in by computer programs to identify the calling program. They are used in API (Application Programming Interface) authentication to control access and ensure security.

12. T-SQL (Transact-SQL):

- T-SQL is Microsoft's proprietary extension of SQL (Structured Query Language) used with Microsoft SQL Server. It includes procedural programming constructs for developing complex business logic.

13. MSSQL Server Management Studio (SSMS):

- SSMS is an integrated environment for managing any SQL infrastructure. It provides tools for configuring, managing, and administering SQL Server instances.

14. ODBC (Open Database Connectivity):

- ODBC is a standard application programming interface (API) for accessing database management systems (DBMS). It allows database-independent communication between a database and an application.

Overview of Sprint Plans

Product owner: Professor Uluagac

Product owner User story: As a system administrator responsible for managing our organization's IoT devices, I want to efficiently keep track of all IoT devices in our inventory, so that I can ensure their proper functioning, security, and maintenance.

This sprint is a highly level overview of what each member will doing throughout the semester

1. Device Registration:

- As a user, should be able to register new IoT devices into the inventory management system.
- Each device should have a unique identifier (e.g., serial number or MAC address) to prevent duplication.
- Product owner should be able to categorize devices by type, such as sensors, actuators, or gateways.

2. Device Information:

- For each registered device, Product owner should be able to capture and store relevant information, including:
 - Device name or label.
 - Manufacturer information.
 - Date of purchase or installation.
 - Location or deployment site.
 - Warranty expiration date, if applicable.
 - Device specifications and capabilities.

3. Device Status Tracking:

- The system should automatically track the status of each device, including whether it is online or offline.
- Product owner should be able to manually update the status of devices as needed (e.g., for maintenance or decommissioning).

4. Alerts and Notifications:

- The system should generate alerts or notifications when devices go offline unexpectedly or when they exhibit abnormal behavior.
- Product owner should receive these alerts via email or within the system's dashboard.

5. Inventory Search and Filter:

- Product owner should be able to search for devices based on various criteria, such as device type, location, or status.
- Filtering options should be available to quickly locate specific subsets of devices.

6. Device Maintenance:

- Product owner should be able to schedule and record maintenance activities for each device.
- The system should provide reminders for scheduled maintenance tasks.

7. Security and Access Control:

- Access to the device inventory management system should be restricted to authorized personnel only.
- User roles and permissions should be defined to control who can add, edit, or delete device records.

8. Device History and Logs:

- The system should maintain a history of changes made to device records, including who made the changes and when.
- Detailed logs of device status changes, maintenance activities, and alerts should be available for auditing purposes.

9. Integration with IoT Platform:

- The system should integrate with our IoT platform to automatically discover and register new devices.
- Device data from the IoT platform should be synchronized with the inventory management system.

10. Reporting and Analytics:

- Product owner should be able to generate reports on the overall health and performance of our IoT device inventory.
- The system should provide analytics to identify trends or patterns in device behavior

System Design

Mobile Application

Login Page – Created using Flutter

Overview of the Login Page Design and Functionality

Design Aesthetics: Created a pleasing login interface using a minimalist design approach for improved clarity and ease of use.

Functionality: Implemented a secure and streamlined login process with email and password authentication

Responsive Design: Utilized Flutter's responsive design features to ensure optimal display on various devices, in this case it is just for Android Smartphones

Utilization of Flutter for Cross-Platform Development

Flutter Framework/UI Components:

Flutter framework and its UI components was used build the login page

Dart Compiler:

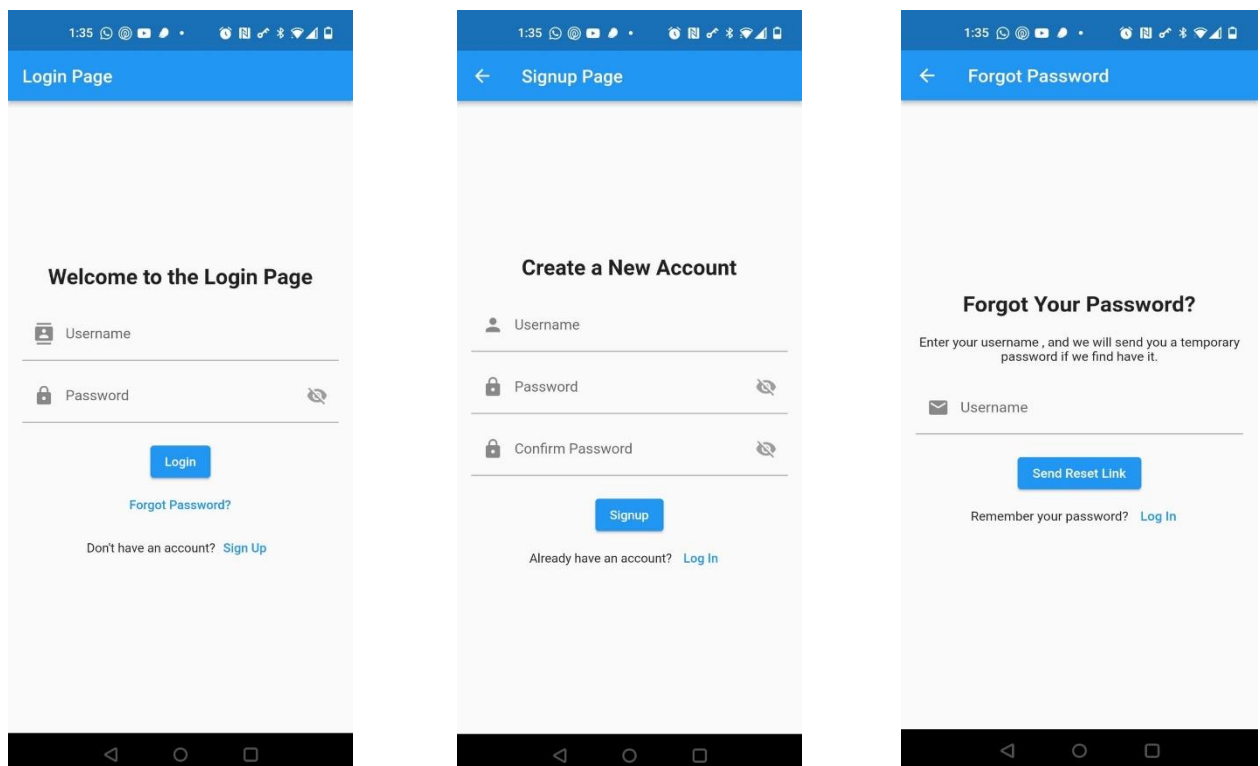
The Dart compiler was responsible for translating Flutter's Dart code into native machine code.

Compiled Code for iOS and Android Platforms:

Illustrates the result of the compilation process, generating native code specific to each platform.

Android Native APIs (Java/Kotlin):

This indicates the interaction of the compiled code with native APIs on each platform.



IoT Device Interface

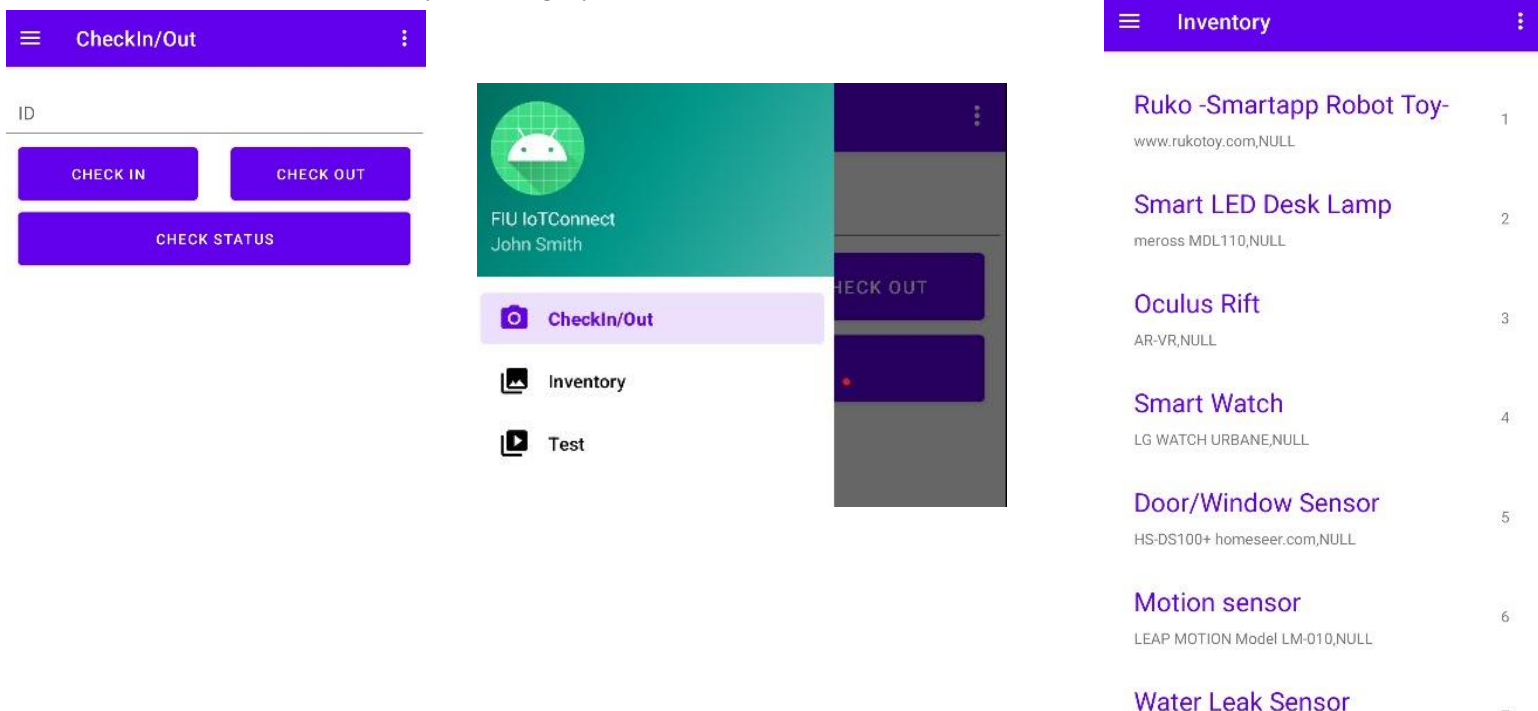
The FIU IoTConnect Mobile Application stands at the forefront of innovation, providing students at Florida International University with a powerful tool to manage IoT devices seamlessly.

Key Features:

1. **User-Friendly Interface:** The application boasts a user-friendly interface, making it easy for students to navigate through the various features. The clean design enhances the overall user experience, ensuring accessibility for users with varying levels of technical expertise.
2. **Device Inventory Management:** Students have access to a comprehensive inventory of available IoT devices. Each device is listed with detailed information, including specifications, availability status, and usage guidelines, aiding students in making informed decisions about their selections.
3. **Check In/Check Out Functionality:** The heart of the application lies in its ability to facilitate seamless check-ins and check-outs of IoT devices. Students can reserve devices for specific durations, and the application sends timely reminders for returning devices before the check-in deadline.

Development Technologies:

- **Android Studio:** The application is developed using Android Studio, the official integrated development environment for Android app development. This ensures optimal performance and compatibility with a wide range of Android devices.
- **Java and XML:** The programming languages used in the development process are Java for the application logic and XML for designing the user interface. This combination allows for robust functionality and a visually appealing layout.
- **Database Integration (T-SQL MSSQL):** The application integrates with a robust database system, such as Microsoft SQL Server (MSSQL), to store and retrieve information about IoT devices, user details, reservations, and quantity data.
- **API Keys and Security Measures:** To ensure secure communication between the mobile application and the server, API keys and encryption protocols are implemented. This guarantees the confidentiality and integrity of user data.



Inventory Management System Website

The IoT Device Management Portal is a robust web application developed to empower administrators at Florida International University (FIU) in tracking and managing IoT devices that students check in check out and quantity

Key Features:

1. Dashboard:

- Upon logging in, administrators are greeted with an intuitive dashboard that provides an overview of the current status of IoT devices.
- Visual representations, such as charts and graphs, offer quick insights into device availability, usage trends, and any pending check-ins/check-outs.

2. Device Inventory Management:

- Administrators can easily add new IoT devices to the inventory or update existing device information.
- Each device entry includes details such as device type, specifications, current availability status, and a comprehensive usage log.

3. Check-In/Check-Out Log:

- A dedicated log keeps track of all device check-ins and check-outs by students.
- Each log entry includes the student's details, the device involved, and timestamps for the transaction.

4. Real-time Quantity Tracking:

- The portal provides real-time tracking of the quantity of each IoT device available in the inventory.
- Administrators can quickly identify devices that are currently checked out and view their expected return dates.

5. User Management:

- Administrators have the ability to manage student accounts, ensuring the accuracy and security of the user database.
- User profiles include details such as student name, contact information, and a transaction history of device check-ins/check-outs.

6. Search and Filter Functionality:

- The portal incorporates robust search and filter features, allowing administrators to quickly locate specific devices or view device usage by selected criteria

Security Measures:

1. Authentication:

- The portal employs a secure authentication system to ensure that only authorized administrators can access the sensitive information.

2. Role-Based Access Control (RBAC):

- RBAC is implemented to define and manage administrator roles, controlling access to specific features and functionalities.

3. Data Encryption:

- All data transmitted between the portal and the server is encrypted to protect against unauthorized access.

About Us

More Info

Members

Students

Search:

Search...

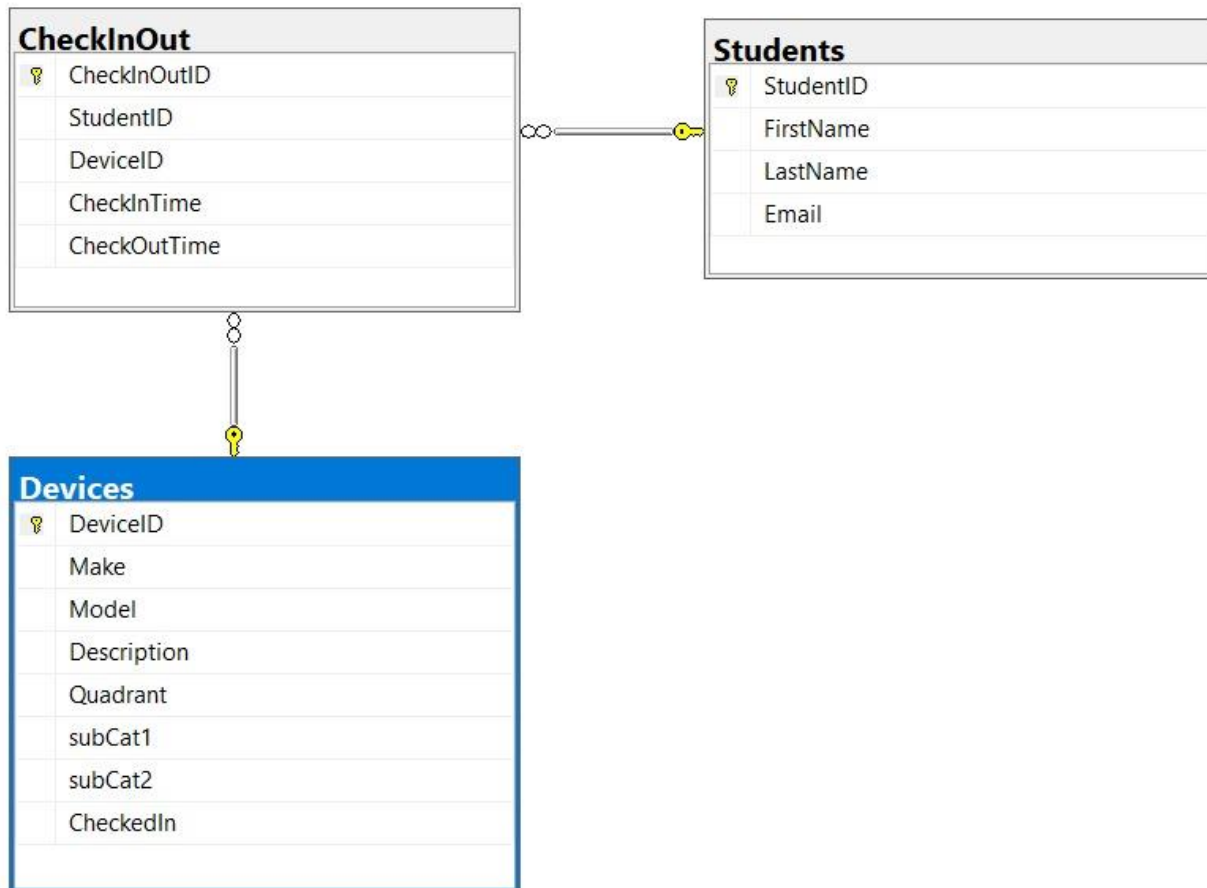
DeviceID	Make	Model	checkedIn	Quadrant	subCat1	subCat2	Description
1	Ruko	Ruko - Smartapp Robot Toy-	False	TOY	Toy	ROBOT	www.rukotoy.com,NULL
2	Meross	Smart LED Desk Lamp	True	AUTOMATION	accessory	LAMP	meross MDL110,NULL
3	Oculus	Oculus Rift	True	COMPUTER	VR	HEAD SET	AR-VR,NULL
4	LG	Smart Watch	True	COMPUTER	WEARABLES	WATCH	LG WATCH URBANE,NULL
5	Homeseer	Door/Window Sensor	True	AUTOMATION	security	Sensor	HS-DS100+ homeseer.com,NULL
6	Leap	Motion sensor	True	AUTOMATION	security	Sensor	LEAP MOTION Model LM-010,NULL
7	SmarthThings	Water Leak Sensor	True	AUTOMATION	security		SmartThings M/N: IM6001-WLP01,16
8	Google	Google Nest Hub	True	AUTOMATION	Smart Speaker	with display	7 inch display 2nd generation ,NULL
9	Amazon	Echo dot - Alexa-	True	AUTOMATION	Smart Speaker		Amazon 3rd generation Bluetooth and WiFi,NULL
10	Amazon	Echo Alexa	True	AUTOMATION	Smart Speaker		Amazon 4rd generation Zigbee Certified WiFi,NULL

Device Inventory Dashboard

Device Name	Status	Actions
Device 1	Available	Check Out Check In
Device 2	Checked Out	Check Out Check In

Appendix A:Database Entity Relation Diagram

This Database Entity Relation Diagram shows the relationship between the tables and how the mobile application works when students check in and or check out the devices



Appendix B:**Sprint 2 Review Meeting Minutes**

Project: FIU IoT Device Inventory Management

Attendees: Zipporah, Khaled

Date: 9.24.2023

Start time: 11:45am

End time: 12:00pm

Note: our team does not have a product owner to accept or reject our user stories, however all user stories that were planned to be done were completed.

After a show and tell presentation, the implementation of the following user stories was accepted by the product owners: All.

Product owner: Professor Uluagac

Product owner User story: As a system administrator responsible for managing our organization's IoT devices, I want to efficiently keep track of all IoT devices in our inventory, so that I can ensure their proper functioning, security, and maintenance.

1. Zipporah: **Device Information – Creating the SQL server Database : Approved**
2. Khaled: **Device Registration** – Finding the NFC tags we will be using for the devices: Approved
3. Michael: **Integration with IoT Platform-** Android Mobile App Creation: Not Approved (still being created)

Sprint 3 Review Meeting Minutes

Project: FIU IoT Device Inventory Management

Attendees: Zipporah, Khaled

Date: 10.08.2023

Start time: 2:30 pm

End time: 3:00pm

1. Zipporah – **Authentication – Not Approved still be worked on with Michael**
2. Khaled: **Website** – Not Approved: Website was created on a public server; therefore it will be created over so it can be more secure

3. Michael: **Graphical User Interface**- Android Mobile App Creation: Approved
4. Michael & Zipporah: **Database** – Approved

Sprint 4 Review Meeting Minutes

Project: FIU IoT Device Inventory Management

Attendees: Zipporah, Khaled

Date:10.08.2023

Start time: 2:30 pm

End time: 3:00pm

1. Zipporah –**Authentication** – **Approved still be worked on with Michael.**
2. Khaled: **Website** – Approved: Website was created on a public server; therefore, it will be created over so it can be more secure and start to secure it to the sql.
3. Michael: **Graphical User Interface**- Android Mobile App Creation: Approved
4. Michael & Zipporah: **Database** – Approved

Sprint 5 Review Meeting Minutes

Project: FIU IoTConnect

Attendees: Zipporah, Khaled

Date:11.05.2023

Start time: 2:30 pm

End time: 3:00pm

1. Zipporah –**Login page** – Approved
 - **Authentication** – Not approved still needs to be tested with Login page and website
2. Khaled: **Website** – Approved: Dashboard has been created for inventory
 - **Authentication** – Not approved still needs to be tested with Login page and website
3. Michael: **Graphical User Interface**- Android Mobile App Creation: Approved
4. Michael & Zipporah: **Database** – Approved

Sprint 6 Review Meeting Minutes

Project: FIU IoTConnect

Attendees: Zipporah, Khaled, Michael

Date: 11.19.2023

Start time: 12:45 pm

End time: 1:00 pm

1. Zipporah – **Login page** – Approved
 - **Authentication** – approved still needs to be tested with Login page and website
2. Khaled: **Website** – Approved: Dashboard has been created for inventory
 - **Authentication** – Approved has been tested with Login page and website
3. Michael: **Graphical User Interface**- Android Mobile App Creation: Approved
4. Michael & Zipporah: **Database** – Approved

Appendix C:

Prerequisites

- **Hardware Requirements**
 - Confirm that the server or hosting environment meets the recommended hardware specifications for the FIU IoTConnect app.
- **Software Requirements**
 - Check for the required software dependencies, including the operating system, web server database server such as MSSQL.
- **User Permissions**
 - Ensure that the user accounts responsible for the installation have the necessary permissions to configure and manage the required components.

Maintenance

- **Regular Backups**
 - Set up a routine backup schedule for both the application files and the database.
 - Test the backup restoration process periodically.
- **User Account Management**
 - Regularly review and update user accounts.
 - Ensure proper access levels for faculty, students, and administrators.

Enhancing FIU IoTConnect User Experience

Potential areas for improvement in the next revisions:

- User Feedback Mechanism
- Enhanced Device Catalog
- Mobile Platform Optimization
- IoT Sensor Integration
- Advanced Reporting and Analytics
- Geolocation Tracking
- Notification Customization
- Offline Functionality Expansion
- Integration with Campus Systems
- Accessibility Features
- Scalability Improvements

Shortcomings/Wishlist

- Limited Device Compatibility
- Dependency on Internet Connectivity
- Limited Reporting and Analytics

Wishlist

- Integration with Campus Systems
- Enhanced User Permissions
- Gamification Elements
- Feedback and Support Mechanism
- Mobile Platform Compatibility
- Enhanced Offline Functionality

References

[Documentation | Android Developers](#)

[Flutter documentation | Flutter](#)

[Create a Simple Website with HTML, CSS, JavaScript \(freecodecamp.org\)](#)

[JSON Web Token Introduction - jwt.io](#)

[Microsoft SQL documentation - SQL Server | Microsoft Learn](#)

[API Reference - OpenAI API](#)

[XML introduction - XML: Extensible Markup Language | MDN \(mozilla.org\)](#)

[Welcome to Flask — Flask Documentation \(3.0.x\) \(palletsprojects.com\)](#)

[Our Documentation | Python.org](#)