

# Tablet Monitor

TCSS 455 Final Report

Group H

Alemneh(Alex) G Shiferaw, Dimitar Kumanov, Leul A Amare



## **Introduction**

At the company where our friend Alemneh works, there is a use of tablets by employees to perform their daily tasks. Employees grab tablets from a tablet station as they need. This leads to some issues such as broken and stolen tablets and lack of accountability. We are trying to solve this problem by creating a website which employees would need to use to check out and return tablets. Additionally, the supervisors would be able to use the website to do inventory each week by seeing which tablets were checked out by whom, as well as how long they were checked out for. We believe that our project can be very useful for a small company that needs to manage the usage of equipment by their employees. The website we built is simple, convenient to use and it does not require any special computer training for employee. This makes our website more flexible and easy to use and avoid cost of training by employers. Additionally, our website will make sure that employees are accountable for the device they use by monitoring their check out and check in time.

## **Objective & Scope Project**

Our goal with this project is to help the company monitor the tablets by using our website and database. This project can be used as a real-world application that would provide value to a company and can be extended further if needed. Our Tablet Monitor website is applicable to many companies in the real world especially, small companies which do not have a lot of resources to develop a large-scale custom solution. Some companies use traditional method to do their inventory for example, checking in and out devices from stations by writing on a paper. This makes it harder to supervisors track down if something is missing from the company and search through the records. Our main goal is to make sure that supervisors can track down if a device has been checked-out and has been returned. We are building our website based on an actual company's needs, we have asked the managers and the supervisor what they are looking for in order to make our website effective and usable as soon as we are done.

## **Relation to Existing Work**

Currently the company makes their employee write down their employee id on paper when they are checking out the tablets. Unfortunately, employees are not usually following that rule. By digitalizing the process of checking out and returning the devices we are hoping that employees are more likely to follow the procedure since the website is simply and easy to use and since the supervisor can easily see if someone has not used it before checking out a device. An additional advantage of using a website

approach is that Supervisors can read see the device records remotely, so they can inspect the employees at their work stations and see if they skipped using the website to check out the device. In the future we can force employees to log in to the tablets with their employee credentials before the tablets are unlocked and can be used. When the employee logs in to the system, it would automatically send a record to the database to record use of the device which would include the unique tablet number, the employee name/id and the current timestamp. This will ensure that employees cannot use the device without producing a record for it and greatly increase the accountability.

## **Main Body of Work & Overview of Architecture**

### **Back-End**

We build our website using the ASP.NET stack using C# for the back-end and MSSQL for DBMS. Most of the functionality of our website was created by using Visual Studio drag and drop, such as dropdown menus that run SQL queries. We used C# in the back-end to perform more complicated operations such as updating multiple relations whenever a device is checked out or returned. We also used C# to redirect the user from one page to another whenever state such as the employee ID needed to be persisted. For example, when an employee types their ID and clicks Log In, our back-end checks whether the ID is numeric, whether that ID is present in our relation, what is the position of the employee, whether that employee has already checked out a tablet and finally redirects the employee to the appropriate web page. An employee that has a 'Team Member' position and has already checked out a device, for example, will be redirected to /UserView\_Return.aspx where they can return that device. Another employee that hasn't checked out a device would be redirected to UserView\_CheckOut. A third employee whose position is Supervisor would be redirected to ManagerView.aspx where they can read the device records and see which employee checked out which device. For all of these redirections, the employee id is captured and added to the redirection link, so the receiving page can recognize which employee is it currently serving and show appropriate welcome screen and functionality.

### **Front-End**

The interface of our site is built on using the latest version of bootstrap, which is 4.0. In order to make the interface very attractive and modern we implemented our idea. Being a proud UW student lead us to a decision to make the client side of the web based on the official UW color, which is purple, gold and metallic gold. To make it more interesting we make an animated background color of our university.

When it comes to the interfaces functionality our site gives a very easy user friendly service. For demonstration purpose the clients can start using their assigned employee id to login to the web. Once they logged in to the web the database will give access based on their employment status. For example, any employee can checkout or return a tablet but only supervisors can see reports of tablet usage.

# Contributions

## Alemneh

This project enabled us to build something from scratch to all the way to final product. I am very happy and gained tons of knowledges by accomplishing our project Tablet Monitor. My contributions for this project started right after the instructor introduced it to our class. I basically came up with the idea of our project and I strongly believe that we have learned a lot from it. I have participated almost on every part of our project mainly on designing and drawing our Relational Model, ER diagram and Normalization. Our project is well organized, we split the work evenly and we had a meeting every week to see or evaluate our progress and to work on it. We have faced a lot of challenges in order to build our database and website, this is due to not having prior experience with C# and connecting database to a website. The in-class activities are very helpful in order to get started on our project; we have used most of the examples to create the database and connect it to webforms. We have used Visual Studio to do most of the tasks for instance, creating the database, SQL queries, dropdownlist and buttons.

## Dimitar

I contributed to the project by organizing and helping split the work. In the beginning of each phase I would set up a shared google doc, post the link in our discussion forum. I would also set up a github so we can share our code, outline the parts needed to be completed, split SQL queries that need to be written so there is accountability in the group and so each of us has a clear idea what they need to complete for the current phase. I also contributed heavily to the SQL schema statements and sample data (especially for Phase 3) and wrote most of the queries used in the back-end. I worked on the back-end part of the website, hooking up the front-end with the queries we wrote, initializing the needed objects for running the queries in C#, etc.. Finally, I completed the BCNF normalization proof, worked on the diagrams (E-R, relational) and contributed several slides for our presentation.

## Leul

Since we start working on this project, I've contributing ideas and implementation to make the project very interesting. Like my other groupmates my participation starts from phase I. On the beginning there was some ambiguity between us about participation since all of us couldn't make time to meet at the same time. After completing the project we decided to meet once a week and split the work as well as sharing ideas to implement the project in a better way. My participation mainly starts on the second phase, i've populated datas to test our queries before we submit it. Additionally, i help my groupmates to design the schima. The biggest part my contribution to the project is building our website, the front-end is fully designed and implemented by me.

## **Future Work**

Since our website is going to be tested by a real company (G2 secure staff), we could expand our efforts based on the feedback we receive. An essential improvement to our project would be to host the website on the cloud since our website runs only on a local machine, but we could push it on the cloud and make it accessible via online to the company. We think that we can expand our project to be installed as an app on each tablet and monitor the login and logout time on the device itself rather than on a single monitoring computer or website. This will be very useful to the company because it can save time for employees making a line in order to check-out/in a tablet from a single or multiple computer. This app idea also gives us a better monitoring system for each individual tablet. For instance, not only monitoring check-out/in tablets but also to check a data usage, hacking tablets in order to access google or work unrelated sites (one of the main problems at G2), location to keep track of the device if it is outside of the workplace zone etc.

Another improvement we can do to our website is add protection against SQL injections. Currently some of the queries we run in the back-end use user input to build up the query string. When we are reading employee ID we validate the input by making sure it's numeric. However, in other places of our website we do not validate or sanitize our input. For example, whenever the employee returns a tablet we use the value of the menu to input the device id. If someone were to use a modified front-end they could send a malicious partial SQL query instead of the device id. This is why we would need to validate all user inputs, possibly removing special characters and/or ensuring only numeric values are allowed for IDs.

## **Conclusion**

In the end of this project we managed to create a website that meets all our initial goals and objectives. This includes providing all the essential functionality of checking out and returning devices by employees and reading the device records by the supervisors. Additionally, we managed to create a simple, friendly user-interface that encourages employees to use our system. Our website is also very lightweight and all operations can be performed with minimal number of interactions: employees can easily log in by only typing their employee number which is typically several digits and they can check out a device or return a device with a single click. This ensures that the company will not lose valuable time in training their employees to use our system or operating and debugging the system.

To sum up, this project gave us a hands on practice for our database course. Despite the limited time kept us from adding more features and functionality we managed the time well and presented something valuable using the knowledge we learned throughout the quarter. Additionally, the project helped us learn new materials in order to achieve our goals. Even though, the courses focus was on building a functional database, the project pushed the limit to go further and build something valuable that will benefit our community and school. By learning database we're able to solving an exciting problem that can easily solvable by a group of three passionate students. We are very thankful for the knowledge we gained throughout the course and the project. It would have been impossible to complete this project without our professors effort to teach us new material beyond database.

## Resources

The following are resources we used throughout our project:

VisualStudio & ASP.NET - We used Visual Studio Community Edition to create our website using ASP.NET with C# for the back-end.

<https://screencast-o-matic.com> - ScreenCast helped us record a video demo of our project

<https://www.draw.io/> - Draw.io was used for the creation of the graphics in this report.

<https://getbootstrap.com/> - The front end of our website is partially designed using bootstrap.

<http://www.washington.edu/brand/graphic-elements/primary-color-palette/> - The RGB color for our website's animation was taken from the UW site

## Appendix A

### E-R Diagrams

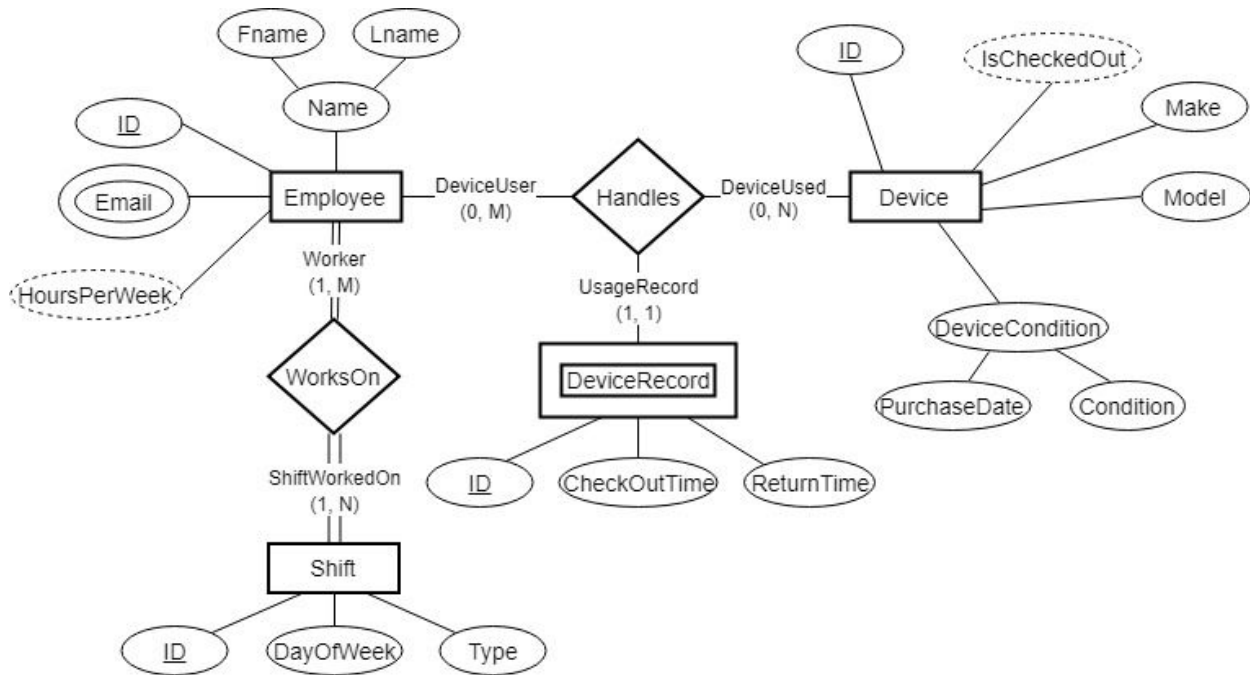


Figure 1. ER-Diagram Before Normalization.

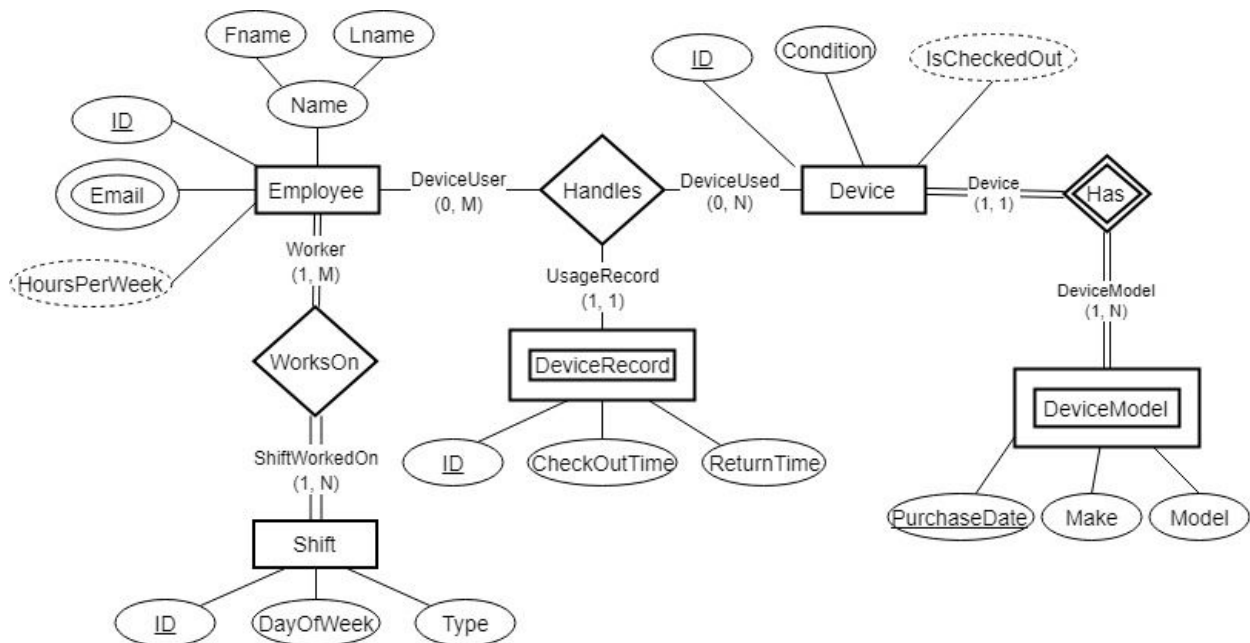


Figure 2. ER-Diagram After Normalization.

## Appendix B\*

### Relations Normalization

\*Appendix B uses the following format:

Table1(Attribute1, Attribute2,..., AttributeN)

#### Functional Dependencies for Table 1

#### Functional Dependencies Before Normalization

Employee (ID, Fname, Lname, Position, Phone, HoursPerWeek)

{ID} → {Fname, Lname, Position, Phone, HoursPerWeek}

EmployeeEmails (EmployeeId, Email)

No functional dependencies.

Shift (ID, DayOfWeek, Type)

{ID} → {DayOfWeek, Type}

WorksOn (EmployeeId, ShiftId)

No functional dependencies.

Device (ID, IsCheckedOut, Make, Model, Condition, PurchaseDate)

{ID} → {IsCheckedOut, Make, Model, Condition, PurchaseDate}

{PurchaseDate} → {Make, Model}

DeviceRecord (ID, CheckOutTime, ReturnTime)

{ID} → {CheckOutTime, ReturnTime}

Handles (EmployeeId, DeviceId, DeviceRecordId)

{DeviceRecordId} → {CheckOutTime, ReturnTime}



## Functional Dependencies After Normalization

### Proof of BCNF

The following are the functional dependencies after optimization. These show that our relational model is in BCNF since all of the determinants in our functional dependencies are primary keys.

Employee (ID, Fname, Lname, Position, Phone, HoursPerWeek)

$\{ID\} \rightarrow \{Fname, Lname, Position, Phone, HoursPerWeek\}$

EmployeeEmails (EmployeeId, Email)

No functional dependencies.

Shift (ID, DayOfWeek, Type)

$\{ID\} \rightarrow \{DayOfWeek, Type\}$

WorksOn (EmployeeId, ShiftId)

No functional dependencies.

Device (ID, IsCheckedOut, Condition, PurchaseDate)

$\{ID\} \rightarrow \{IsCheckedOut, Condition, DescriptionOfCondition, PurchaseDate\}$

DeviceModel (PurchaseDate, Make, Model)

$\{PurchaseDate\} \rightarrow \{Make, Model\}$

DeviceRecord (ID, CheckOutTime, ReturnTime)

$\{ID\} \rightarrow \{CheckOutTime, ReturnTime\}$

Handles (EmployeeId, DeviceId, DeviceRecordId)

$\{DeviceRecordId\} \rightarrow \{CheckOutTime, ReturnTime\}$

## Appendix C

### Relational Diagram

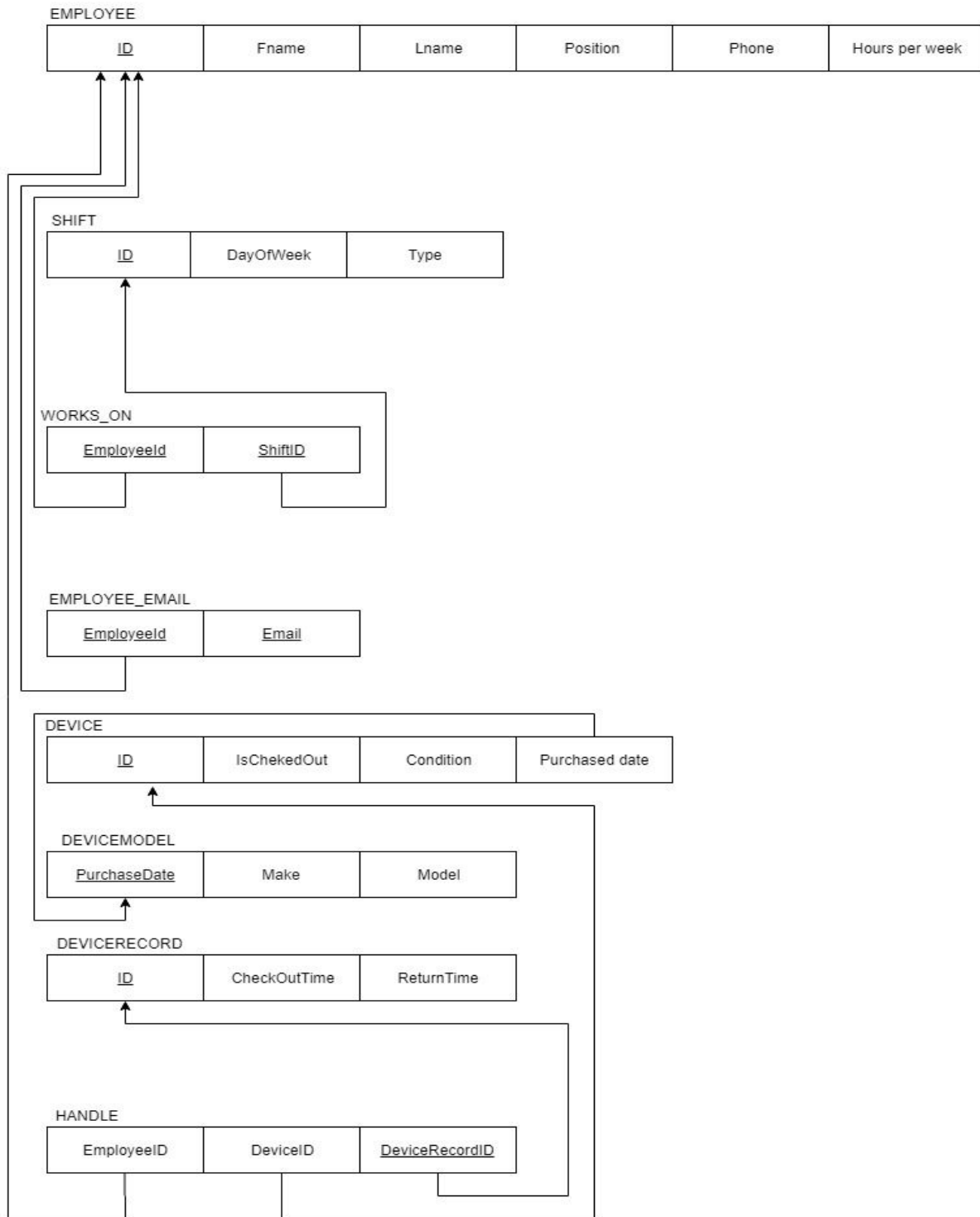


Figure 3. Relational Schema Diagram After Normalization.

## Appendix D

### Functional SQL Queries

```
/* QUERY 4
* Objective: Determine if employee has checked out a device (if so which device)
given an employee id.
* Arguments:
    {0} - employee id
* Used In: LogIn.aspx to determine whether the employee has checked out a tablet
already. This is used to decide if the employee should be redirected to
UserView_CheckOut.aspx or UserView_Return.aspx.
*/
Select Device_Handle.DeviceId, Device_Record.CheckOutTime
From Device_Handle, Device_Record
Where {0} = Device_Handle.EmployeeId AND
    Device_Handle.DeviceRecordId = Device_Record.Id AND
    Device_Record.ReturnTime IS NULL;
```

Figure 4. SQL Query 4 - Determine the device checked out by an employee.

```
/* QUERY 5 (multiple queries, but treated as a single transaction)
* Objective: Check out a device by an employee. This involves inserting rows and
making changes in several relations.
* Arguments:
    {0} - the next available device record id, which is calculated by adding one to
the first query
    {1} - the current date/time formatted as yyyyMMdd HH:mm:ss tt (tt is AM/PM)
    {2} - the ID of the employee checking out the device
    {3} - the ID of the device being checked out
* Used In: UserView_CheckOut.aspx to check out the chosen device by the logged in
user.
*/
{0} = (Select MAX(ID) From Device_Record; + 1)

Insert Into Device_Record(ID, CheckOutTime, ReturnTime)
Values({0}, {1}, NULL);
Insert Into Device_Handle
Values ({2}, {3}, {0});
Update Device
Set IsCheckedOut = 1
Where Device.ID = {3};
```

Figure 5. SQL Query 5 - Check out a device for an employee.

```

/* QUERY 9 (multiple queries, but treated as a single transaction)
* Objective: Return a device that was previously checked out by an employee.
* Arguments:
    {0} - the ID of the device being returned. (see QUERY 7)
    {1} - the device record ID corresponding to this device being checked out by this
    employee. (see QUERY 8)
    {2} - the current date/time formatted as yyyyMMdd HH:mm:ss tt (tt is AM/PM).
* Used In: UserView_CheckOut.aspx to return the device currently checked out by the
logged in employee.
*/
Update Device
Set IsCheckedOut = 0
Where Device.Id = {0};
Update Device_Record
Set ReturnTime = {2}
Where Device_Record.Id = {1};

```

Figure 7. SQL Query 9 - Return a device that was checked out by an employee.

```

/* QUERY 10
* Objective: Show all device records, including the ID of the device being checked
out, the name, id and phone of the employee checking out the device, the time the
device was checked out and returned (or empty if still checked out). Additionally,
results are sorted by check out time in order to show the most recent device usages
first.
* Arguments: None
* Used In: ManagerView.aspx to display all device records for the manager to inspect
and potentially contact employees that have not returned their devices.
*/
Select Device_Handle.DeviceId, EMPLOYEE.Fname, EMPLOYEE.Lname,
Employee.Phone as PhoneNumber, EMPLOYEE.Id as EmployeeID,
Device_Record.CheckOutTime, Device_Record.ReturnTime
From EMPLOYEE, Device_Handle, Device_Record
Where Device_Record.Id = Device_Handle.DeviceRecordId AND
EMPLOYEE.Id = Device_Handle.EmployeeId
Order By Device_Record.CheckOutTime DESC;

```

Figure 8. SQL Query 10 - Show all device records for the manager to inspect.

# Appendix E

## Functional Web Interface

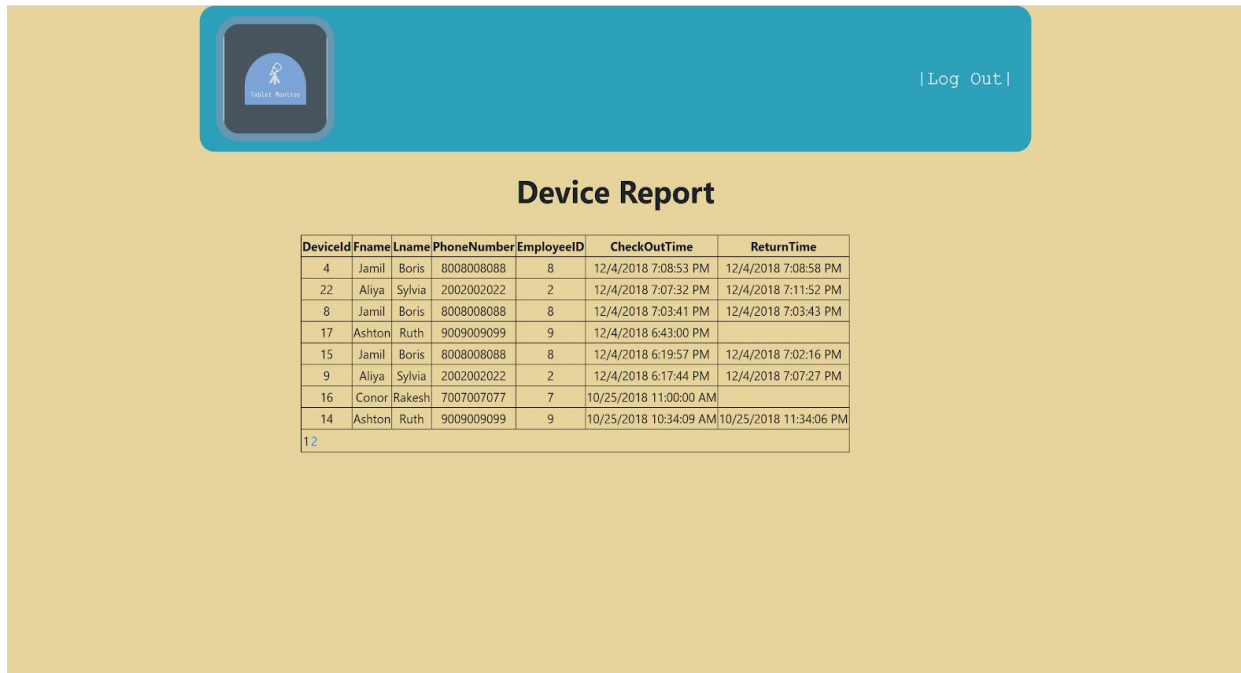


Figure 4. Screenshot a device report for the manager.

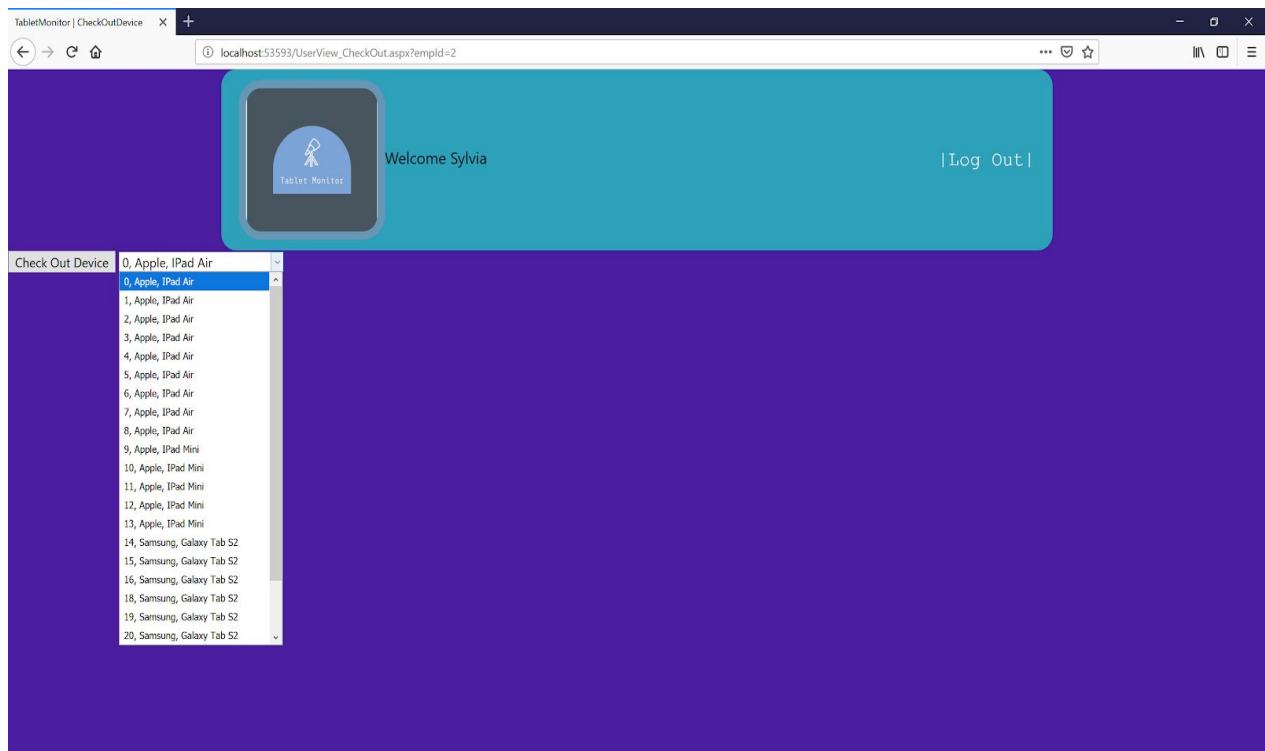


Figure 5. Screenshot of a an employee logged in to checkout a tablet.



