Department of Computer Science and Engineering
The University of Texas at Arlington

Team: Team Argus

Project: PixelSense Secure Transfer - Lynx

Team Members:
Shamikul Amin
Brandon Deen
Brian Hasty
Keyur Patel
Seth Skocelas

Last Updated: 3/4/2015 9:03 PM

Table of Contents

Document Revision History

Revision Number	Revision Date	Description	Rationale
1.0	11/20/2014	Final Submission	Approved by team, submitting for approval from sponsor and supervisor
0.9	11/15/2014	Revised Draft	Added SDK sections, improvements based on feedback
0.8	11/5/2014	Peer Group Submission	Edits made after feedback from Sponsor
0.5	10/8/2014	Initial Draft	Needed to be created

List of Figures

List of Tables

1. Product Concept

The PixelSense Secure Transfer system is a tablet case with a series of sensors allowing secure transmission between the PixelSense table and the device (the Lynx) connected to the tablet. The benefit of this transfer method is that communication cannot be intercepted due to physical limitation, thus making the connection secure. Included with the Lynx will be an SDK that will allow users to develop their own applications for both the tablet and the PixelSense table in order to use it however they see fit.

For demonstration purposes, our group will be developing a casino game on the PixelSense table, and our secure transfer system attached to a tablet will serve as a poker chip counter, securely storing your chips from table to table.

1.1 Purpose and Use

This device is being developed in order to provide a secure connection to transfer data between the PixelSense table and whatever is connected to our Lynx device. When the Lynx is placed on the table with compatible software loaded, the table will be able to recognize the Lynx, and be able transfer data when needed.

With the application we're developing for the table, a casino game, the Lynx will provide a way to securely store casino chips, and also be used as a secondary screen when playing the game. For example, if you were playing poker, you can display the cards on the tablet where no other player can see them as opposed to showing them on the table.

Since we will be developing an SDK for the Lynx, it can be used for many other purposes in tandem with the PixelSense table. Below are some ideas that could be implemented using the SDK we create:

- A health information tracker that stores information on the device, and can only be displayed when the device is placed on the PixelSense table. A doctor can then add or remove information using the table as he/she sees fit.
- An enhanced chess game where the transfer device serves as a chess piece that can store information regarding player habits.
- A degree plan tracker that can store advising information about a student, and when the device is placed on the table, the student's degree plan and academic information would be displayed for the advisor, which the advisor could edit as he/she sees fit.

1.2 Intended Audience

The intended audience of the Lynx is anyone who has the PixelSense table and the need to transfer small amounts of information to and from the table securely. Using our SDK, they can develop applications that take advantage of this secure method of transferring data.

The intended audience for the demonstration application we are developing would be any casino interested in using the PixelSense table to play their games as opposed to traditional, non-interactive tables.

2. Product Description and Functional Overview

This product is a multi-component, anti-cheating system for casino games. The product includes a Microsoft PixelSense, and the Lynx device. The core purpose of this product is to transfer data back and forth between the PixelSense and the Lynx optically, to avoid data interception and modifications. From this core idea, it stemmed to developing an anti-cheating system. The system will work as follows:

1) A Lynx will be checked out to a customer.
2) The customer places Lynx on PixelSense to start playing.
3) The customer chooses a game and starts playing.
4) Score and money are stored onto the Lynx.
5) The customer finishes playing and returns the Lynx to cash out.
6) Lynx is returned and cashes the customer out.

The initial game included is blackjack. The Lynx will connect to an external device such as a tablet or computer to view contents. The system will look like the following:

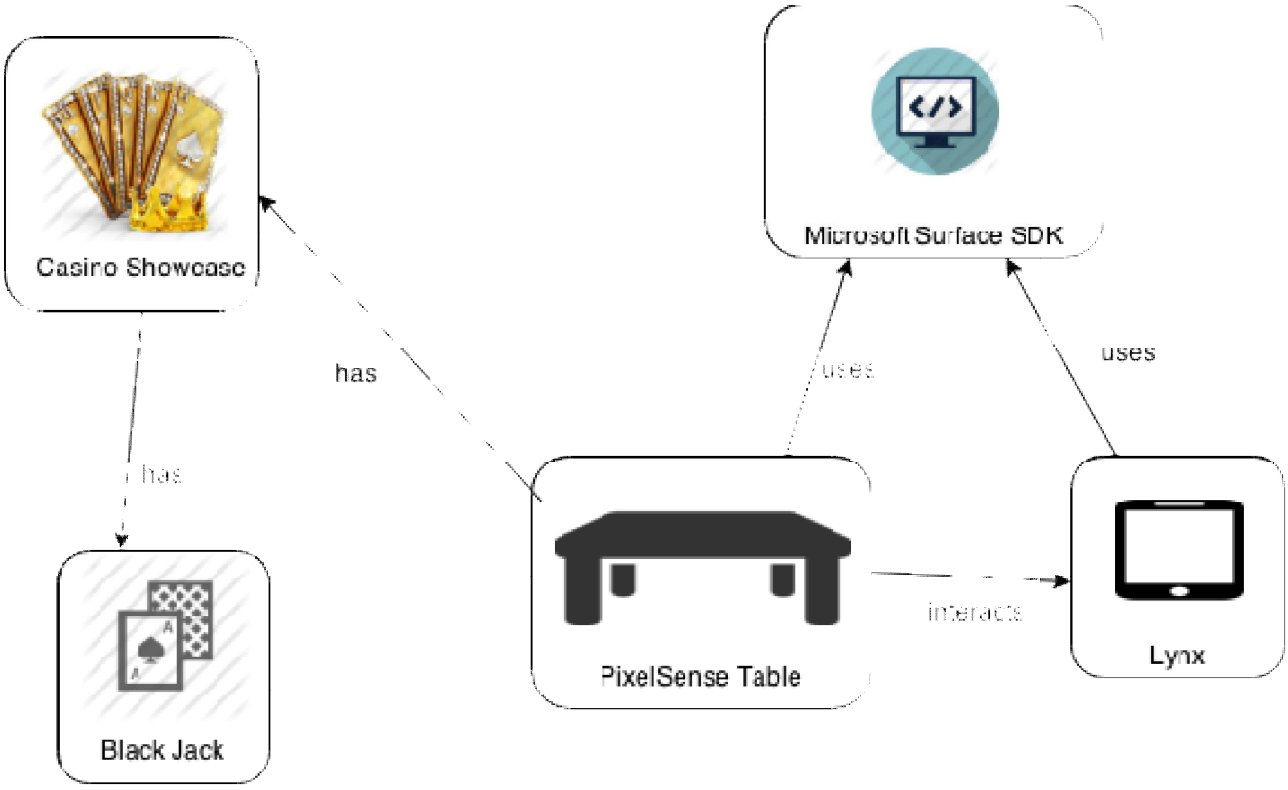

Figure 2-1: PixelSense Secure Transfer System Overview

2.1 Features and Functions

First, we have the ability to view data externally on a tablet or a computer. The Lynx will be able to communicate optically with the Surface table. Going onto the PixelSense, the table should recognize the Lynx and distinguish it from other objects placed on the table. The table should also be able to transmit

and receive data at a respectable speed and not keep the customer waiting. The Lynx should be self-contained. The table should communicate everything locally, optically or through a physical cable. There should be no data going over the internet or a wireless network, to avoid data interception.

2.2 External Inputs and Outputs

The Lynx does not receive direct input from the user itself, but rather indirectly from the PixelSense. The Lynx also is able to send input to the PixelSense, forming a complete 2-way connection. The PixelSense itself is able to communicate with the Lynx for various tasks. The Lynx will also have a serial output for the ability to display data on an external screen, tablet or computer.

Name:	Description	Use
Authentication	Lynx should authenticate itself with the table.	Find if an official Lynx is being used.
Data Verification	Lynx should be able to connect to an external computer or tablet.	Verify that the data does match
Optical Transmission	Data must be transferred optically between the Lynx and table	Transfer of betting information

Table 2-1 Product Communication

2.3 Product Interfaces

2.3.1 This is the home screen of the Casino Showcase Application. A user can select a game he/she wants to play by just tapping on each.

Figure 2-2 Homepage

2.3.2 This is the Lynx screen where the user is asked to put the Lynx on the table to start the game.

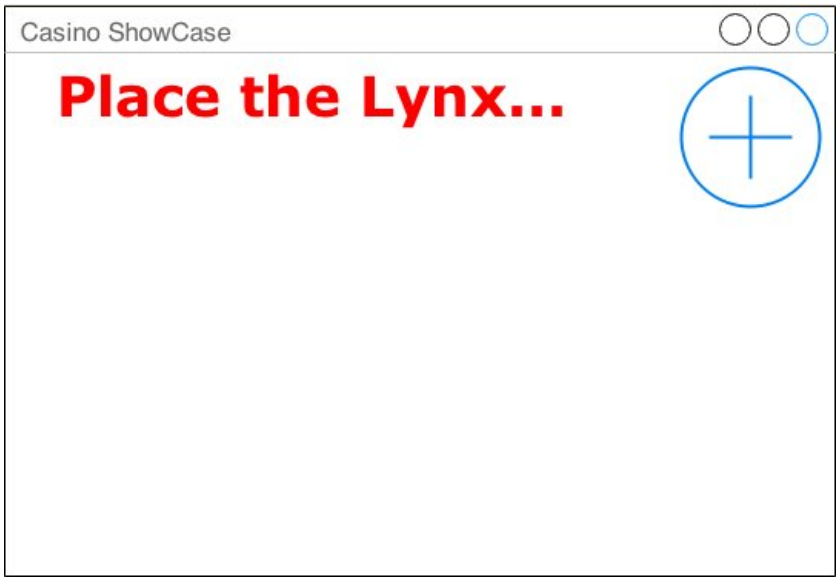

Figure 2-3 Waiting for placement of Lynx

2.3.3 This is the Lynx verification screen where the system is identifying the Lynx and preparing the game.

Figure 2-4 Identifying Lynx

2.3.4 This is the Lynx Access confirmation screen, which is shown when the Lynx is successfully identified by system and has loaded all data for the game.

Figure 2-5 Access Granted

2.3.4 This is the actual Blackjack casino Game. The User sees this when he/she has successfully completed the above steps. The Screen shows a user's name, chip count, the Blackjack game environment, and an exit button to quit the game.

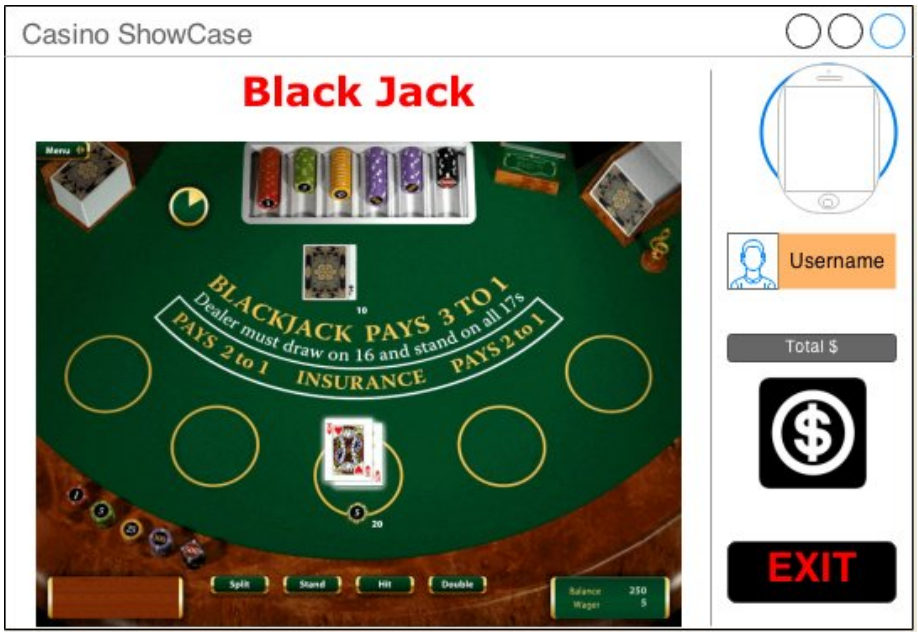

Figure 2-6 Playing Blackjack

2.3.4 If a user disconnects Lynx from the table, the system will display a pop up message asking the user if the Lynx was removed intentionally. User is allowed to put Lynx back to resume a game.

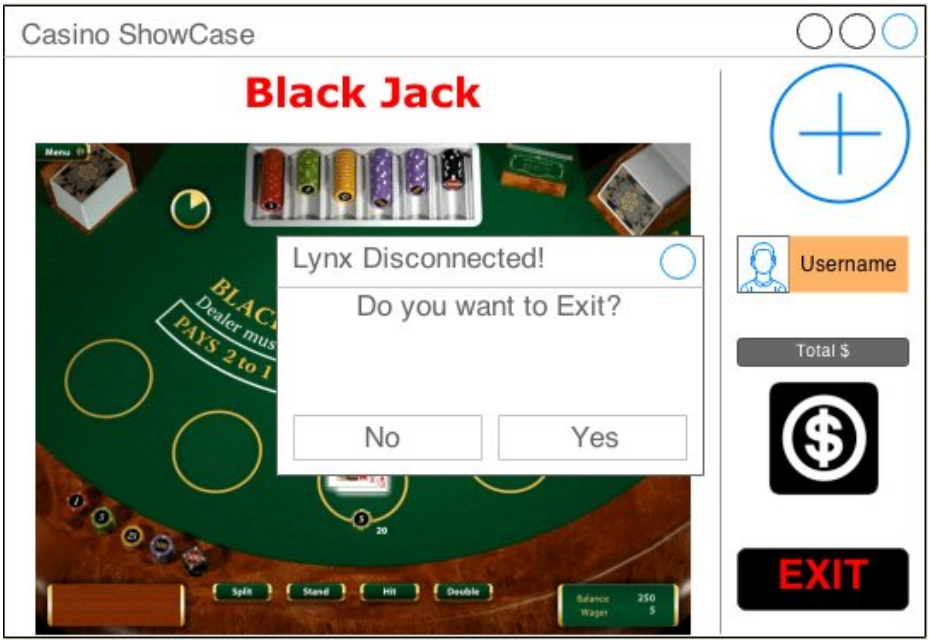

Figure 2-7 Removed Lynx from table

3. Customer Requirements

The following section contains the customer requirements agreed upon by Team Argus and Dr. Gergely Zaruba. These requirements cover the creation of the device, the SDK, and the casino software associated with this project.

3.1 The product shall be able to send data optically

3.1.1 Description: The product must be able to send specifically timed flashes of light as bits of information and send them in a way a receiving unit can interpret it.

3.1.2 Source: The requirement is from Professor Gergely Zaruba.

3.1.3 Constraints: The position of the device on the table may cause writing problems. The protocol that handles writing should be software agnostic; not relying on specific programs outside of the provided SDK in order for it to work.

3.1.4 Standards: None

3.1.5 Priority: 5 (Very High)

3.2 The product shall be able to read data optically

3.2.1 Description: The product must be able to receive specifically timed flashes of light as bits of information and interpret them correctly, storing the data if necessary.

3.2.2 Source: The requirement is from Professor Gergely Zaruba.

3.2.3 Constraints: The position of the device on the table may cause reading problems. The protocol that handles reading should be software agnostic; not relying on specific programs outside of the provided SDK in order for it to work.

3.2.4 Standards: None

3.2.5 Priority: 5 (Very High)

3.3 All work done by the product involving optical communication will be compiled into a well-documented library

3.3.1 Description: A suite of functions designed to handle optical communication between capable hardware will be available in a portable SDK library. The details of this requirement is elaborated on in the next section.

3.3.2 Source: The requirement is from Professor Gergely Zaruba.

3.3.3 Constraints: None

3.3.4 Standards: None

3.3.5 Priority: 5 (Very High)

3.4 The device must have a serial port to transfer data collected

3.4.1 Description: The Lynx will have a serial port to transfer data optically collected to the Android device connected.

3.4.2 Source: The requirement is from Professor Gergely Zaruba.

3.4.3 Constraints: Transfer will only work with the tablet specification we develop for.

3.4.4 Standards: None

3.4.5 Priority: 5 (Very High)

3.5 The device and table's communication should be error tolerant

3.5.1 Description: All data sent should be received intact, and be identical to the original message. The Lynx should detect errors in information received.

3.5.2 Source: The requirement is from Professor Gergely Zaruba.

3.5.3 Constraints: None

3.5.4 Standards: None

3.5.5 Priority: 4 (High)

3.6 Communication between both devices must sustain a reliable connection

3.6.1 Description: All data sent should be received intact, and be identical to the original message. The protocol will have a system to detect missing or corrupt data, and replace the missing pieces with the correct information.

3.6.2 Source: The requirement is from Professor Gergely Zaruba.

3.6.3 Constraints: None

3.6.4 Standards: None

3.6.5 Priority: 4 (High)

3.7 Software will be built to showcase the optical transfer protocol and the built device

3.7.1 Description: A program will be supplied to demonstrate the capabilities of the optical communication SDK, and Lynx. The program will play a game of Blackjack, with the device holding chip information.

3.7.2 Source: The requirement is from Professor Gergely Zaruba.

3.7.3 Constraints: None

3.7.4 Standards: None

3.7.5 Priority: 5 (Very High)

4. SDK Requirements

The following section contains the customer requirements agreed upon by Team Argus and Dr. Gergely Zaruba that cover the creation of the SDK associated with this project. Unless specified otherwise, the functions mentioned in these requirements will be a part of all three subsystems of our project. (The Lynx device, the PixelSense table, and Android platform)

4.1 The system will support Android API 17 and Surface 2.0 platforms

4.1.1 Description: An SDK will be provided for both Android API 17 (4.2 and up) and Microsoft Surface 2.0 (PixelSense) platforms. An SDK will also be provided for the Lynx device itself.

4.1.2 Source: Team Argus

4.1.3 Constraints: None

4.1.4 Standards: None

4.1.5 Priority: 5 (Very High)

4.2 The system will support customizable light sequences

4.2.1 Description: The Device, Android and PixelSense SDK will provide functions that gives ability to customize the sequence of light transmitted and to set what happens when those light sequences occurs.

4.2.2 Source: Team Argus

4.2.3 Constraints: None

4.2.4 Standards: None

4.2.5 Priority: 3 (Moderate)

4.3 The system shall allow error correction when transferring data

4.3.1 Description: The SDK will support some form of error correction when data is submitted.

4.3.2 Source: Dr. Zaruba

4.3.3 Constraints: None

4.3.4 Standards: None

4.3.5 Priority: 4 (High)

4.4 The system shall allow the expansion of the array

4.4.1 Description: The Device and PixelSense SDK functions provided will allow the user to set the size of the array of transmitters they want to use and support that functionality.

4.4.2 Source: Dr. Zaruba

4.4.3 Constraints: The array must be square, meaning n x n (e.g. 4x4).

4.4.4 Standards: None

4.4.5 Priority: 2 (Low)

4.5 The system shall support a 4x4 array of transmitters and receivers

4.5.1 Description: The Device and PixelSense SDK will provide support for a 4x4 array to transmit and receive data to and from the Surface table.

4.5.2 Source: Dr. Zaruba

4.5.3 Constraints: None

4.5.4 Standards: None

4.5.5 Priority: 4 (High)

4.6 The system shall authenticate the Lynx device when plugged into the tablet

4.6.1 Description: The Android and Device SDK will provide function(s) that will allow the Android application to authenticate that a valid Lynx device is plugged into the Android tablet.

4.6.2 Source: Dr. Zaruba

4.6.3 Constraints: None

4.6.4 Standards: None

4.6.5 Priority: 4 (High)

4.7 The system shall allow the Lynx to send and receive data

4.7.1 Description: The Device and Android SDK will provide functions that will allow the Android application developed with it to send and receive data using the Lynx device.

4.7.2 Source: Team Argus

4.7.3 Constraints: None

4.7.4 Standards: None

4.7.5 Priority: 5 (Very High)

4.8 The system shall allow the PixelSense to send and receive data

4.8.1 Description: The PixelSense SDK will provide functions that will allow the PixelSense application developed with it to send to and receive data from the Lynx device.

4.8.2 Source: Team Argus

4.8.3 Constraints: None

4.8.4 Standards: None

4.8.5 Priority: 5 (Very High)

4.9 The system shall be able to determine the orientation of the Lynx

4.9.1 Description: The PixelSense SDK functions will provide the ability to determine how the Lynx is oriented on the Surface table. These functions can leverage existing functionality present in the development kit for PixelSense.

4.9.2 Source: Dr. Zaruba

4.9.3 Constraints: None

4.9.4 Standards: None

4.9.5 Priority: 5 (Very High)

4.10 The system shall notify the Android device if the Lynx is on the table

4.10.1 Description: The Device and Android SDK will provide information to the user via the Android tablet connected to the Lynx when the Lynx is on the Surface table and ready to transmit or receive data.

4.10.2 Source: Dr. Zaruba

4.10.3 Constraints: None

4.10.4 Standards: None

4.10.5 Priority: 5 (Very High)

4.11 The system shall notify the table that the Lynx is on it

4.11.1 Description: The Device and PixelSense SDK will provide information to the user via the Surface table that the Lynx is on the Surface table and ready to transmit or receive data.

4.11.2 Source: Dr. Zaruba

4.11.3 Constraints: None

4.11.4 Standards: None

4.11.5 Priority: 5 (Very High)

4.12 The system shall authenticate that a valid Lynx device is placed on the PixelSense table.

4.12.1 Description: The PixelSense SDK will provide functions to authenticate that a valid Lynx device is on the PixelSense table and is capable of sending and receiving data.

4.12.2 Source: Dr. Zaruba

4.12.3 Constraints: None

4.12.4 Standards: None

4.12.5 Priority: 5 (Very High)

4.13 The system shall support multiple devices

4.13.1 Description: The PixelSense SDK will provide functions to detect and support multiple Lynx devices on the table.

4.13.2 Source: Dr. Zaruba

4.13.3 Constraints: None

4.13.4 Standards: None

4.13.5 Priority: 3 (Moderate)

5. Packaging Requirements

In this section, we'll cover the requirements for how we will deliver our product to the customer. The Secure Transfer device will be a self-contained entity (the Lynx) already assembled for the user and ready for use. Since this will be used with a tablet, the Lynx will need to be able to attach/detach from the tablet. Our SDK will be loaded on a CD, which can be used on Windows computers to develop for both Android and Surface SDKs. The Casino software we develop will be available on a USB drive to be loaded on a PixelSense Table.

5.1 Attachable to Tablet

5.1.1 Description: The Lynx must be attachable and detachable physically from the tablet it is connected to. This is referring to a mechanism so the Lynx can physically stay on the tablet it is connected to, not the serial connection with the cord plugged into the tablet.

5.1.2 Source: Dr. Zaruba

5.1.3 Constraints: The Lynx will only be attachable/detachable for the one tablet we choose to develop on. (Only referring to the mechanism to keep the Lynx physically attached to the tablet, not the serial connection with the cord plugged into the tablet.

5.1.4 Standards: None

5.1.5 Priority: 3 (Normal)

5.2 Fully Assembled

5.2.1 Description: The Lynx will be assembled and ready for use.

5.2.2 Source: Dr. Zaruba

5.2.3 Constraints: None

5.2.4 Standards: None

5.2.5 Priority: 4 (High)

5.3 Lynx as a Tablet Case

5.3.1 Description: The Lynx will serve as a tablet case for the tablet we chose to develop for.

5.3.2 Source: Dr. Zaruba

5.3.3 Constraints: Case will only fit the tablet we choose to develop for.

5.3.4 Standards: None

5.3.5 Priority: 2 (Low)

5.4 SDK on USB Flash Drive

5.4.1 Description: The Lynx' SDK for the PixelSense table, the Secure Transfer Device, and the connected device will be provided on a USB flash drive for use on Windows.

5.4.2 Source: Dr. Zaruba

5.4.3 Constraints: None

5.4.4 Standards: None

5.4.5 Priority: 4 (High)

5.5 PixelSense Casino Software

5.5.1 Description: The PixelSense Casino Software will be provided on a USB Flash Drive and the software can be installed on the table.

5.5.2 Source: Dr. Zaruba

5.5.3 Constraints: None

5.5.4 Standards: None

5.5.5 Priority: 4 (High**)**

5.6 PC Casino Companion Software

5.6.1 Description: The PC Casino Companion Software will be packaged as an executable file on an USB Flash Drive that can be used on compatible Windows PCs.

5.6.2 Source: Dr. Zaruba

5.6.3 Constraints: None

5.6.4 Standards: None

5.6.5 Priority: 4 (High)

5.7 Android Casino Software

5.7.1 Description: The Android Casino Software will be packaged in an APK on an USB Flash Drive that can be installed on compatible Android tablets.

5.7.2 Source: Dr. Zaruba

5.7.3 Constraints: None

5.7.4 Standards: None

5.7.5 Priority: 4 (High**)**

6. Performance Requirements

In this section, we will go over the necessary performance requirements for the entire system as well as the individual components. The different aspects of this product mean that we will have different performance requirements over each component. The components that will be covered are as follows: optical transfer, data translation, connection times, authentication times, software boot times and data read/write times.

6.1 Minimum Data Transfer Rate

6.1.1 Description: Data transfer must be consistent at a minimum of 200 bits/sec

6.1.2 Source: Dr. Zaruba

6.1.3 Constraints: PixelSense refresh rate is between 30-60Hz, this must be kept in mind

6.1.4 Standards: None

6.1.5 Priority: 4 (High)

6.2 Data Translation

6.2.1 Description: Converting optical data to binary data for later readability should be at most 1 second.

6.2.2 Source: General Consensus

6.2.3 Constraints: None

6.2.4 Standards: None

6.2.5 Priority: 3 (Normal)

6.3 Authentication Time

6.3.1 Description: The Lynx should not take too long to authenticate its integrity with the PixelSense. Time should at most 5 seconds.

6.3.2 Source: Dr. Zaruba

6.3.3 Constraints: Must be 2 way authentication.

6.3.4 Standards: None

6.3.5 Priority: 3 (Normal)

6.4 Software Boot Times

6.4.1 Description: Software should not take a long time to boot the developed application and load data. Ideally this should be less than 10 seconds.

6.4.2 Source: General Consensus

6.4.3 Constraints: PixelSense processing power as well as concurrent PixelSense background processes

6.4.4 Standards: None

6.4.5 Priority: 4 (High)

6.5 Data Read/Write Times

6.5.1 Description: Reading and writing entire datasets to the Lynx data through serial should be at most 1 second. This accounts for all writing delays to the Lynx flash/EPROM.

6.5.2 Source: General Consensus

6.5.3 Constraints: Depends on how fast our chosen microcontroller is.

6.5.4 Standards: None

6.5.5 Priority: 4 (High)

6.6 Battery Life

6.6.1 Description: Battery Life on the Lynx should last about half a casino day which is about 8 hours.

6.6.2 Source: General Consensus

6.6.3 Constraints: None

6.6.4 Standards: None

6.7.5 Priority: 4 (High)

6.7 Overall Connection Times

6.7.1 Description: Connection times to the Lynx, tablet and DB should take no longer than 5 seconds.

6.7.2 Source: Dr. Zaruba

6.7.3 Constraints: Network speed and hardware transmission limitations

6.7.4 Standards: None

6.7.5 Priority: 3 (Moderate)

7. Safety Requirements

In this section we will discuss some basic safety requirements that the product should contain to protect the consumer. The requirements will include physical safety aspects of the Lynx.

7.1 No sharp edges

7.1.1 Description: The Lynx must not have any sharp edges

7.1.2 Source: General Consensus

7.1.3 Constraints: None

7.1.4 Standards: None

7.1.5 Priority: 4 (High)

7.2 Protective Case around Device

7.2.1 Description: Lynx should have a protective case to prevent exposure to liquids

7.2.2 Source: General Consensus

7.2.3 Constraints: None

7.2.4 Standards: None

7.2.5 Priority: 4 (High)

8. Maintenance and Support Requirements

This section contains maintenance and support requirements for the PixelSense Secure Transfer system. It explains how to maintain a fully functioning system to the user. These requirements must be met to make the system maintenance and upgrade easy and effective.

8.1 User Manual

8.1.1 Description: A user manual shall include a step by step guide to use the SDK with device & software on Surface Table.

8.1.2 Source: PixelSense Secure Transfer

8.1.3 Constraints: User shall be required to have knowledge of operating a touch based Windows 7 Operating System.

8.1.4 Standards: None

8.1.5 Priority: 4 (High)

8.2 Software Installation

8.2.1 Description: Software shall be provided to interact between device and PixelSense Table. Software shall not require additional permission to install software. Software shall work on any Windows 7 Operating System on the PixelSense Hardware.

8.2.2 Source: PixelSense Secure Transfer

8.2.3 Constraints: User shall be required to have knowledge of operating touch based Windows 7 Operating System.

8.2.4 Standards: None

8.2.5 Priority: 4 (High)

8.3 Source Code / SDK

8.3.1 Description: Source code shall be provided to the customer as final deliverable. Basic Software Development Kit with usage guide shall be provided in order to use it further for PixelSense Development.

8.3.2 Source: PixelSense Secure Transfer

8.3.3 Constraints: None

8.3.4 Standards: None

8.3.5 Priority: 3 (Moderate)

8.4 Hardware Support

8.4.1 Description: Device shall be ready to use with PixelSense Secure transfer system. Device shall include Android Operating System with PixelSense Application. If required, customers can replace the device with the same application in future.

8.4.2 Source: PixelSense Secure Transfer

8.4.3 Constraints: A user shall know how to install application on android platform.

8.4.4 Standards: None

8.4.5 Priority: 4 (High)

9. Other Requirements

This section describes implicit requirements for the PixelSense Secure Transfer to be complete and not previously listed.

9.1 American English Standard

9.1.1 Description: The PixelSense Secure Transfer shall use American English as the default language for any text or audio speech.

9.1.2 Source: Team Argus

9.1.3 Constraints: None

9.1.4 Standards: None

9.1.5 Priority: 4 (High)

9.2 User Friendly Interface

9.2.1 Description: The PixelSense Secure Transfer shall have a user interface that can be learned in under 2 hours, and includes guidance to aid the user during use of the application.

9.2.2 Source: Team Argus

9.2.3 Constraints: None

9.2.4 Standards: None

9.2.5 Priority: 4 (High)

10. Acceptance Criteria

Listed below are the acceptance criteria that must be met in order for our project to be accepted by our customer. Due to the customer's main focus being providing the Lynx with an SDK to develop for it, the casino software specifics are not listed within the acceptance criteria.

10.1 The device can securely transfer information to the PixelSense table using an array of sensors

10.1.1 Requirement(s) addressed:

Requirement 3.1 - Send Data Optically.

Requirement 4.7 - The system shall allow the Lynx to send and receive data

Requirement 4.8 - The system shall allow the PixelSense table to send and receive data

10.1.2 Verification Procedure: The customer will use the software provided for the PixelSense table and the tablet connected to the Lynx to verify that the table can receive information from the tablet connected to the Lynx. All other forms of communications will be turned off on both the tablet and the table to insure that the Lynx alone was transmitting the information.

10.2 The device can securely receive information from the PixelSense table using an array of sensors

10.2.1 Requirement(s) addressed:

Requirement 3.2 - Read Data Optically.

10.2.2 Verification Procedure: The customer will use the software provided for the PixelSense table and the tablet connected to the Lynx to verify that the table can send information to the tablet connected to the Lynx. All other forms of communications will be turned off on both the tablet and the table to insure that the Lynx alone was transmitting the information.

10.3 An SDK is provided that can be used to program for the device created

10.3.1 Requirement(s) addressed:

Requirement 3.3 - All work done by the product involving optical communication should be compiled into a well-documented library

Requirement 4.1 - The system will support Android API 17 and Surface 2.0 platforms

10.3.2 Verification Procedure: The customer will be given the code used to create the secure connection between the table and the Lynx, as well as any documentation created for the Lynx, and verify whether or not the resources provided are sufficient enough to create an application using the Lynx and the table.

10.4 The transfer rate of the Lynx is at least 200 bits/sec

10.4.1 Requirement(s) addressed:

Requirement 5.1 – Minimum Transfer Rate

10.4.2 Verification Procedure: The customer will use the software developed for the Lynx and the table, as well as any tools the customer deems necessary, to verify the Lynx satisfies the minimum data transfer throughput requirements.

10.5 The Lynx has a port that can be used to connect to a tablet

10.5.1 Requirement(s) addressed:

Requirement 3.4 – Device must have a serial port.

10.5.2 Verification Procedure: The customer will use the port on the Lynx and verify that when an Android tablet is connected to it, the Android tablet can transfer data through the Lynx using the software provided.

10.6 The software is provided for the PixelSense table that demonstrates the connection between the Lynx and the table

10.6.1 Requirement(s) addressed:

Requirement 3.7 – Software will be built to show optical transfer protocol.

10.6.2 Verification Procedure: The customer will use the casino software provided and verify that is demonstrates the secure connection between the Lynx (with the tablet connected to it) and the PixelSense table. The customer will do this using the software provided for the PixelSense table, as well as the software for the tablet connected to the secure transfer Lynx.

10.7 Software developed for the PixelSense can detect the orientation of the Lynx

10.7.1 Requirement(s) addressed:

Requirement 4.9 - The system shall be able to determine the orientation of the Lynx

10.7.2 Verification Procedure: The customer will use verify that there is a function within the PixelSense SDK that allows software developed with it to track the orientation of the Lynx device.

10.8 Lynx Software developed for the PixelSense, Device, and Android Platform can detect when a Lynx device is present

10.8.1 Requirement(s) addressed:

Requirement 4.10 - The system shall notify the table that the Lynx is on it

Requirement 4.11 - The system shall authenticate that a valid Lynx device is placed on the PixelSense table.

10.8.2 Verification Procedure: The customer will use the casino software provided and verify that is demonstrates the secure connection between the Lynx (with the tablet connected to it) and the PixelSense table. The customer will do this using the software provided for the PixelSense table, as well as the software for the tablet connected to the secure transfer Lynx.

11. Use Cases

This section shows possible uses of the Casino Showcase Application and android companion application with PixelSense and Android SDK through usage of the Lynx device, with an emphasis on cases that demonstrate the functionality of the SDK. “This Use Case Begins With” has been abbreviated to “TUCBW” and “This Use Case Ends With” with “TUCEW” for this section.

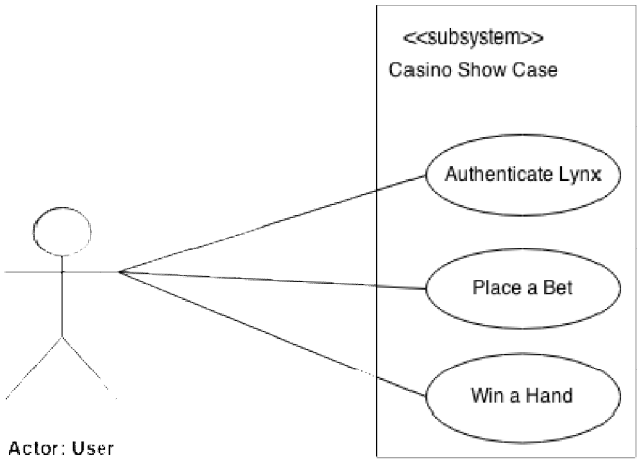

Figure 11-1 Casino Showcase Subsystem

11.1 Authenticate Lynx

11.1.1 **Scenario:** The user places the Lynx device on the PixelSense surface table. The PixelSense surface SDK recognizes the shape and orientation of the device and sends a request for user ID to Lynx via a sequence of lights. Lynx recognizes the sequence and translates it. Once Lynx has translated the sequence, Lynx will forward the request to Android SDK. The Android SDK responds back to Lynx with numerical user ID. Lynx converts the number into light sequences and signals it back to PixelSense Table. The PixelSense surface SDK recognizes the sequence and translates the data back to a number for the verification process. The PixelSense SDK verifies the Lynx, if it is verified, the user sees the confirmation with message “Lynx successfully authenticated.” on PixelSense Table.

11.1.1.2 **Actor:** User

11.1.1.3 **TUCBW** User places Lynx device on PixelSense surface table.

11.1.1.4 **TUCEW** User sees the confirmation message “Lynx successfully authenticated” after Lynx has been verified by PixelSense SDK.

11.2 Place a Bet

11.2.1 **Scenario:** A user selects bet amount on PixelSense table and presses the “Place Bet” button. PixelSense SDK sends request with bet amount from demo game to Lynx by converting request to sequence of lights. Lynx catches the light sequence and translates it. It then deducts the requested amount from the Lynx. If the amount is being properly deducted, it sends Boolean value back to PixelSense SDK via Android SDK through light sequence. Surface SDK recognizes the sequence and translate it back to Boolean value. If the Boolean value is true, user sees number being displayed on demo game under bet amount as the bet is placed successfully else if the Boolean value is false, user sees the “You don't have enough funds to place bet”.

11.2.2 **Actor:** User

11.2.3 **TUCBW** User selects bet amount on PixelSense table and press “Place Bet” button.

11.2.4 **TUCEW** User sees bet amount being displayed on demo game under bet amount after the amount is being deducted from Lynx.

11.3 Winning a Hand

11.3.1 **Scenario:** A user sees the message on demo game “You won!” The demo game sends the winning amount to PixelSense SDK. The PixelSense SDK sends a request with the winning amount to Lynx by converting request to sequence of lights. Lynx catches the light sequence and translates it. It then adds the requested amount to the Lynx. If the amount is being properly added, it sends Boolean value back to PixelSense SDK via Android SDK through light sequence. The Surface SDK recognizes the sequence and translates it back to Boolean value. If the Boolean value is true, user sees number being displayed on demo game “Ready for another bet”.

11.3.2 **Actor:** User

11.3.3 **TUCBW** User selects bet amount on PixelSense table and press “Place Bet” button.

11.3.4 **TUCEW** User sees number being displayed on demo game “Ready for another bet” after the amount is being added to Lynx.

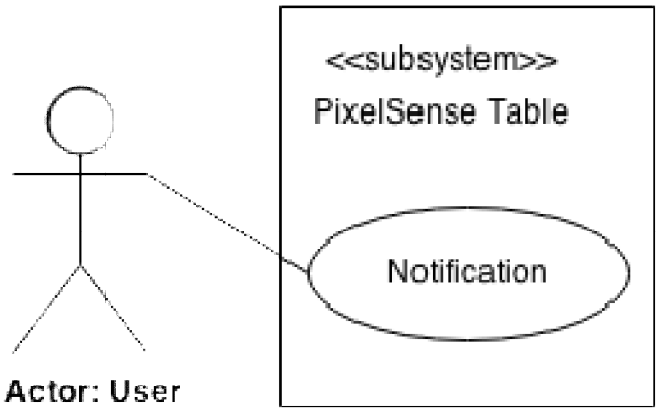

Figure 11-2 PixelSense Table Subsystem

11.4 Notification

11.4.1 Scenario: A user places a Lynx device on PixelSense table. The PixelSense SDK recognizes the shape and orientation of the device and sends sample data to Lynx via sequence of lights. If Lynx is connected to a tablet, Lynx will follow the authentication process (Use Case 11.1), or else Lynx will respond back with same light sequence. Once the PixelSense SDK receives some light sequence from Lynx, it will send a notification request to Android SDK that Lynx is on table. The Android SDK generates a notification from the request. The user sees the notification on android device stating that Lynx is placed on the table.

11.4.2 Actor: User

11.4.3 TUCBW User places Lynx device on PixelSense surface table.

11.4.4 TUCEW User sees the notification on android device stating that Lynx is placed on the table.

12. Feasibility Assessment

This section analyzes the feasibility of the PixelSense Secure Transfer project, by covering the following analyses: scope analysis, research, technical analysis, cost analysis, resource analysis and schedule analysis.

12.1 Scope Analysis

The scope of work for all requirements critical to this project, as well as prototyping the requirements by the deadline is both reasonable and feasible. This assessment is based off of the research performed by the team and discussions with the project sponsor. The expectations are that the bulk of the scope of work will be comprised of three requirements, which are prioritized in the order that follows: the construction of the Lynx, the software development kit for the Lynx and the casino application software, prioritized in that order. Furthermore, the addition of high priority criteria only slightly adds to the scope of work, while maintaining the high probability of completion.

12.2 Research

This section includes two subsections, the research that has already been accomplished and the research that needs to be done, which allows a better understanding of the feasibility of the project.

Research completed:

- Hardware (AT Mega 328 chip, Arduino)
- Optimal programming language (C#)
- Possible game engines (Unity, XNA)

Research to be completed:

- Game engines (Will Unity or XNA suit our needs? Are they necessary?)
- Hardware (sensors, diodes, tablets, chips)
- C# (Gain more experience in this language)
- Game development (for Blackjack)
- PixelSense application development (get familiar with SDK)

12.3 Technical Analysis

The project will be a simulated casino table where the Lynx simulates the user's collection of chips. The Lynx will, when placed on the PixelSense, interact with the casino application. While the user plays a game of Blackjack. The funds that are won or lost during the process are actively added or removed from the Lynx. The hardware will consist of an array of sensors and receivers to send and receive data with the PixelSense, a chip to drive the Lynx, and a serial port to connect it to a tablet. Only two of the five team members have experience in this area. This means that as a team, we will need to research optimal ways to build the Lynx and how to send and receive data to and from the PixelSense. The software will consist of a PixelSense casino application that includes blackjack, a mobile application that allows users to connect the Lynx to a tablet, and an SDK to enable others to develop similar products. No one on the team has any experience with the PixelSense or game development, but all members have experience in mobile application development. The team will also need to research the following: how the PixelSense will receive the data from the Lynx and send data back to the Lynx, as well as, how to develop a casino application for the PixelSense.

12.4 Cost Analysis

The team is budgeted $800 for all equipment needed to build a prototype, this excludes the purchase of the PixelSense which belongs to the department and is well over our budget. After performing research we have concluded that we will be within the specified budget. The PixelSense Secure Transfer is a simulated casino where all chips are held virtually on the Lynx that interacts with the PixelSense. The majority of the budget will be the cost needed to build this Lynx and a tablet to interface with the Lynx through the serial port.. We have determined that we will not need to purchase any other software. Table 12-1 shows the breakdown of the cost analysis.

Description	Cost
Lynx ● Custom Minimal Control Board ● Array of Transmitters and Receivers ● Printed Circuit Board ● USB-Serial Breakout Board	$100 - 150.00
Tablet with serial port	$75 - 100.00
Microsoft PixelSense	$0.00
Total Cost:	$175 - 250.00

Table 12-1 Cost Analysis

12.5 Resource Analysis

Team Argus consists of two computer scientists, two software engineers, and one computer engineer. Our computer engineer and one of our computer scientists have extensive knowledge and experience in hardware and sensors. All members have knowledge in a variety of programming languages and object-oriented programming. The software engineers will be able to analyze the risks associated with this project, utilize any design patterns and architect the required software. The team lead will be able to keep the team organized and on track through the use of MS Project and management skills.

12.6 Schedule Analysis

Jones First Order Estimation will be used for schedule estimation, which allows us to get a rough estimate of the total time it may take to complete the project. Below inputs and outputs have been categorized into five subsections and ranked by difficulty (low, medium, high) based upon the number of Data Element Types (DET) and the number of File Types Referenced (FTR). Function points were derived for each component of the project using the International Function Point User Group (IFPUG) tables.

Program Characteristic	Complexity	Function Points
Number of Inputs	-	-
● To Tablet from Lynx	Low	3
● To PixelSense from Lynx	High	6
● To Lynx from PixelSense	Medium	4
● To PixelSense from User	Medium	4
Number of Outputs	-	-
● Lynx to Tablet	Low	4
● Lynx to PixelSense	High	7
● PixelSense to Lynx	Medium	5
● Administrator PC to Lynx	Low	4
Inquiries	-	-
● Authenticate Lynx	Medium	4
Logical Internal Files	-	-
● Casino funds tracking	Medium	10
● Casino Application (Blackjack, Homepage…)	Medium	10
External Interface Files	-	-
● Lynx information to Tablet	Low	5
● Casino Application on PixelSense	Medium	7
● Lynx information on PixelSense	Medium	7
Total Unadjusted Function Points		80

Table 12-2 Unadjusted Function Points

After determining our unadjusted total we can calculate our Value Adjustment Factor based off of the 14 different System Characteristics. The degree of influence is ranked for each characteristic from 0-5.

General System Characteristic	Influence (0-5)
Data Communications	3
Distributed Data Processing	0
Performance	5
Heavily Used Configuration	1
Transaction Rate	5
On-Line Data Entry	0
End-User Efficiency	3
On-Line Update	0
Complex Processing	1
Reusability	5
Installation Ease	3
Operational Ease	3
Multiple Sites	0
Facilitate Change	3
Total	32

Table 12-3 System Characteristic

Using the Total above we can find the Influence factor by doing the following calculation:

$$\textit{Influence Factor} = .65 + (32 / 100) = .97$$

Using the Influence factor and the Total unadjusted Function Points found previously (80) we get:

$$\textit{Total Adjusted Function Points} = 80 * .97 = 77.6$$

Through the use of Jones' First Order Estimation Practice the following estimate for our schedule was made using this calculation:

$$(\text{Total Adjusted Function Points})^{\text{Case Exponent}} = \#\text{ of months}$$

General Application	Exponent	Estimated Schedule (months)
Best Case	.43	6.49
Average Case	.45	7.08
Worst Case	.48	8.07
Modified Worst Case	.48	7.52

Table 12-4 Schedule Estimation

As of 07 October 2014 there is roughly 7.5 months remaining to finish our project and deliver a prototype. Our estimates put us at 7.08 months until completion, which means the team is currently on track, but gives little room for error. Modifying the worst case scenario to reflect the removal of low priority requirements, we arrive at 7.52 months which gives a slightly more feasible project in the worst case. We will have continual oversight on this throughout the development process. In conclusion, the project is currently feasible; as development begins the team will remain vigilant in determining any possible risks that may delay the project more than necessary. If anything arises then the number of games available on the Casino Application will decrease or the expected graphical quality of those games will be lowered in order to ensure the project is still feasible under the current deadlines.

13. Future Items

Listed below are project general requirements that could be achieved, but cannot be implemented in the time or budget allotted under current estimates. Rather than reject good requirement, they are listed here for use in the next generation of the product.

13. 1 Add more games to the Casino Program

13.1.1 Description: Before the first betting of a round, give the players an option to changes games. Other games to consider adding are: Poker and variations), Roulette, Baccarat, or Slots

13.1.2 Source: Team Argus

13.1.2 Constraint: Only if time permits.

13.1.4 Standards: none

13.1.5 Priority: 2 (Low)

13. 2 Add Graphical Fidelity to Blackjack (or other games)

13.2.1 Description: Instead of rudimentary test graphics and menus, the program could expand to use the range the PixelSense table offers, with creative ways to display and interact with chip count and virtual cards. Possibly allow for customizable table images and card faces.

13.2.2 Source: Team Argus

13.2.2 Constraint: Only if time permits.

13.2.4 Standards: none

13.2.5 Priority: 2 (Low)

13. 3 Iterate on the Hardware

13.3.1 Description: Through implicit learning with building and researching optical devices, time may be allotted at the end of the project for hardware improvements, to boost responsiveness, capacity, or battery longevity to the device.

13.3.2 Source: Team Argus

13.3.2 Constraint: Only if time permits.

13.3.4 Standards: none

13.3.5 Priority: 2 (Low)