

University of Alabama in Huntsville

LOUIS

Honors Capstone Projects and Theses

Honors College

4-24-2019

Theater Ticket Pickup Kiosk App

Bradley Sawyer Johnson

Follow this and additional works at: <https://louis.uah.edu/honors-capstones>

Recommended Citation

Johnson, Bradley Sawyer, "Theater Ticket Pickup Kiosk App" (2019). *Honors Capstone Projects and Theses*. 420.

<https://louis.uah.edu/honors-capstones/420>

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.

Theater Ticket Pickup Kiosk App

by

Bradley Sawyer Johnson

An Honors Capstone

submitted in partial fulfillment of the requirements

for the Honors Diploma

to

The Honors College

of

The University of Alabama in Huntsville

April 24, 2019

Honors Capstone Director: Dr. Rick Coleman

Lecturer of Computer Science

Bradley Johnson 4/30/2019
Student (signature) Date

Rick Coleman 4-30-2019
Director (signature) Date

H.S. Nungesser/LE 5/1/19
Department Chair (signature) Date

[Signature] 5/1/19
Honors College Dean (signature) Date



Honors College
Frank Franz Hall
+1 (256) 824-6450 (voice)
+1 (256) 824-7339 (fax)
honors@uah.edu

Honors Thesis Copyright Permission

This form must be signed by the student and submitted as a bound part of the thesis.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Bradley Johnson

Student Name (printed)

Bradley Johnson

Student Signature

4/30/2019

Date

Theater Ticket Pickup Kiosk App

by

Bradley Sawyer Johnson

An Honors Capstone

submitted in partial fulfillment of the requirements

for the Honors Diploma

to

The Honors College

of

The University of Alabama in Huntsville

April 24, 2019

Honors Capstone Director: Dr. Rick Coleman

Lecturer of Computer Science

Student (signature) Date

Director (signature) Date

Department Chair (signature) Date

Honors College Dean (signature) Date



Honors College
Frank Franz Hall
+1 (256) 824-6450 (voice)
+1 (256) 824-7339 (fax)
honors@uah.edu

Honors Thesis Copyright Permission

This form must be signed by the student and submitted as a bound part of the thesis.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Student Name (printed)

Student Signature

Date

Table of Contents

Abstract.....	3
Introduction	4
Chapter 1: Design	5
Chapter 2: Implementation.....	6
Chapter 3: Results	8
Conclusion	9

Abstract

Theaters often utilize specialty-made kiosk systems capable of printing out tickets. These kiosks often include expensive, custom hardware that would be expensive to repair or replace. This project attempts to create a replacement for specialty-made kiosk systems using a normal printer and a tablet.

To achieve this, I plan on connecting to the server for the theater ticket manager created in a senior project. The server will be modified to send a barcode through an email. The kiosk app will be able to use the camera to scan a barcode and print a ticket. This should create a convenient system that is easy to use and simple to setup.

By connecting to existing server code, the scope of this project is greatly reduced since modification to the server code should be minimal. In addition, external libraries will be able to handle the more difficult tasks of generating barcodes, sending emails, and scanning barcodes. These external libraries should make it so that the application is affordable to develop for companies/groups with smaller budgets.

Introduction

Technology introduces a lot of conveniences for us, but some of these conveniences come with a large price tag. However, recent leaps in mobile technology have made tablet hardware cheaper and more accessible with plenty of libraries to provide additional functionality. By taking advantage of this, we can create a simple and affordable system to bring these conveniences to the customers of smaller businesses.

The system I've chosen to create will be a ticket pickup kiosk designed to be used at local Huntsville theaters. Large places like movie theaters have the conveniences of online ticket ordering and fast ticket pickup through the use of a website and a kiosk. My kiosk system would connect to a pre-existing ticket ordering website that was created separately for the theater groups. So, the same conveniences would be provided with cheap hardware and simple software making this affordable for small, local theaters.

Chapter 1: Design

The initial design started with what software and hardware would be used for the kiosk application. It needed to be something cheap and convenient. It had to be able to scan and print. It also had to be relatively easy to write software for. Given these factors, I decided to develop the application as an Android app.

Android devices normally have a camera. This can be used for the barcodes. Android also has a printing API that can be used to print the tickets. In addition, the large market for Android apps means that there are a lot of development resources and documentation. Also, I have experience with writing Android apps and own an Android device that can be used for prototyping. A production kiosk could be a generic android tablet mounted and connected to a printer.

So, there will be two components to the system, the ticket website service and the ticket kiosk. The ticket web service will run using NodeJS and will provide three different services. It will allow the customer to order the ticket. It will email the customer a barcode. Lastly, the website will send the ticket info to the kiosk upon request. The kiosk app will scan barcodes using the tablet's camera and print the tickets using a standard home printer

Then, I researched for an external library for scanning barcodes. I found one called Mobile Vision that was created by Google. It had the capability to detect barcodes, faces, and text; however, I would only need the barcode recognition functionality. I also needed a way to generate the barcodes. So, I researched for barcode generation libraries that worked with NodeJS and found a great one called JSBarcode and Canvas. Together, these were able to generate and render barcodes on the server that could be attached to emails. From here, all I had to do was implement these libraries and write applications to use them.

Chapter 2: Implementation

I started by creating a blank Android project. From there, I designed a very basic user interface consisting of a label, a preview window for the camera, a text field to enter a ticket ID into, and a submit button. These provided all of the necessary user interaction. Due to the relative simplicity and shallowness of the user interface, no view patterns were necessary. However, I did create a view model to play with Android view model library. A view model normally sits between the view and model of an application and dictates the appearance and actions of the view. I used the view model to handle requests to the web server and the printer. So, the submit button simply needed a listener that called on the view model when the button was pressed.

Pressing submit retrieves the id from the textbox, validates the id, requests the ticket's information from the server, parses the information into objects, renders them as html, and passes the html to Android's printer service. The most complicated part of this process is the message parsing and rendering. The parsing takes in a JSONObject object and returns a list of Ticket objects. This is simply a lot of calls to the json library to parse out all of the fields from the json and adding them to the ticket object. After the object is finished being created, it gets rendered by a rendering object. The rendering object takes in a list of tickets and a listener to be notified when the html is rendered for printing. The rendering is just a lot of text concatenation to create the correct html text. The final html text is then rendered out as a web page by the Android API and sent to the listener.

Once the printing capability was successfully added, the barcode functionality was implemented next. At first, I tested to see if an email viewer would execute JavaScript since JSBarcode could create the barcode in Javascript. However, I quickly found out that most email viewers would not, so all of my barcodes would have to be created before the email was sent. So, I added the canvas library as a

dependency to my NodeJS server. This allowed the JSBarcode to render a barcode on a server-side canvas. The rendered barcode was then converted to an image and embedded into a new email. This email was then sent using the existing email API used by the ticket manager.

With a large portion of the app implemented, only the scanning functionality was left to add. This proved to be challenging due to the miniscule amount of documentation compared of a complex API. After struggling for several hours, I downloaded a sample app created by Google and tested it. After seeing that it worked, I copied over the core code from that sample app. Once the necessary alterations were made, the scanning was tested successfully. Then all of the functionality was connected to create a working kiosk app.

Chapter 3: Results

Despite some setbacks, the overall results were promising. The only problems encountered were my lack of printer ink and my struggle with Mobile Visions. The printer ink situation was swiftly resolved by a trip to the store. My issues with Mobile Visions was solved by finding a sample application that demonstrated how to use the API. With those problems solved, the app worked well. Only one or two bugs were found and fixed. The app recognized barcodes really well and really quickly. The app can be set into a kiosk mode using a feature in Android called screen pinning which makes it so the notifications drawer and the navigation bar don't appear.

Some small bugs appeared but were soon removed. There are some improvements that could be made. For instance, when the printer API is called to print a ticket, it takes you to a separate app to select the printer and the options. From what I could tell, the printer API doesn't allow you to specify the options beforehand so that the user isn't redirected away from the app. I think I could communicate with the printer directly over the network as I've seen this done by someone else, but that would not be easy to do and could create compatibility issues. I think the current implementation is acceptable but less convenient. In addition, the tickets don't print on separate pieces of paper, so I added a dotted line to signify that they could be cut as a compromise. Given all of this, I'd say the app is usable and successful.

Conclusion

Many larger businesses are able to provide convenient services to their customers through custom, purpose-built hardware. This hardware is expensive to create and maintain and the software can be complex. To allow this same convenience for customers of smaller businesses, common hardware can be used to provide similar results. With a rise in mobile applications and devices, a tablet can provide a decent user experience for less. By taking advantage of Android APIs and libraries, a custom ticket kiosk system can be created to provide the convenience of ticket pickup to users using just a tablet and a printer.