Team Salsa:   
Porting the Tanga iOS Application to Android

CIS 440: Systems Design and E-commerce  
Arizona State University  
December 3, 2015

**Table of Contents**

[Abstract 3](#_Toc436861863)

[The Client: Tanga 4](#_Toc436861864)

[The Problem 4](#_Toc436861865)

[Team Salsa’s Solution 4](#_Toc436861866)

[The Team 5](#_Toc436861867)

[Functionality of Application (Intended) - iOS 8](#_Toc436861868)

[Functionality of Application (Actual) - Android 14](#_Toc436861869)

[Business Issues: Communication 21](#_Toc436861870)

[Technology Issues: Browser-Based Application 22](#_Toc436861871)

[Technology Issues: Native Application 23](#_Toc436861872)

[Technology Stack 24](#_Toc436861873)

[Testing Procedures 24](#_Toc436861874)

[Testing Data Analysis 25](#_Toc436861875)

[Next Steps 26](#_Toc436861876)

# Abstract

From Arizona State University’s CIS 440 capstone class, five seniors joined together to form Team Salsa. The team members include Manoj Panikkar, Kevin Windhorst, Mario Padilla, Valina Gewargis, and John Read. Team Salsa chose to work with Tanga, an e-commerce company based out of Arizona, to develop a native mobile application. Tanga developed an existing application on the iOS platform using Swift, Apple’s new programming language. However, Tanga needed the app ported over to Android. There was no existing code for the android application; the team would have to start development from the ground up mirroring the functionality of the iOS app.

From the entire list of features and functionality of the current iOS app, Tanga listed a few as the initial requirements that Team Salsa should prioritize during development. These requirements stated that the application should be developed natively for Android, should include a container to display navigation items and content on the main screen (including a menu bar at the top), and should also include product descriptions with pictures and details in a list view. Finally, the application should render the product and checkout pages in a mobile web view.

The Software Development Kit used for the development and testing of this mobile application was Android Studio. Team members also had to learn Java, as the entire application would be coded using this programming language. Basic mobile development knowledge, familiarity with APIs, and basic computer science concepts need to be incorporated into this project.

Early on, Team Salsa encountered barriers in communication with Tanga. It took a strong initiative from the team members to acquire the feedback and information needed to complete this project successfully. Some technology-based issues included building a browser-based version of the application as the first iteration, as well as coding a mobile app in Java when none of the team members had any prior experience in either.

Tanga’s mobile application for Android, with a majority of the initial requested features, has been completed. Moving forward, the application and the source code will be handed over to Tanga’s developers. They can choose to either release the app on Android’s mobile market, Google Play, or develop the app further before release.

# The Client: Tanga

Tanga is an e-commerce company that provides various products online at cheaper prices. The company was founded in 2006, and has established an impressive online presence since then. Since 2006, they have shipped over 4 million orders, saved shoppers $400 million off list prices, and served 2 million unique customers. Tanga offers every customer the same promise: “If you ever have problems with your order, we’ll do everything we can do to make it right.” They are a small company, employing only 25 people, and aim to treat customers “like family”. Tanga first started as the founders selling extra board game stock online at a discount. It grew when their community added different types of games. Tanga started adding other types of products, and then moved towards the vast portfolio of items available today. In September of 2014, Tanga was named #18 on the Fastest Growing Private Company in Arizona by the Phoenix Business Journal.

# The Problem

The problem presented was fairly straightforward: Tanga had an existing application for iOS that they needed ported over to Android. The iOS app was first created using Objective-C, and then recoded entirely into Swift, Apple’s new programming language. Our team was instructed that there was no existing code for the Android application; all Android application features would need to be developed from the ground-up. Fortunately, Tanga had a functional back-end component that already provides information to their iOS application. The menu navigation items and the content on the application main screen are sourced from a REST API that delivers JSON, which is parsed and rendered in standard mobile components. All of the product pages, checkout pages, and other app information are provided as mobile web views from Tanga’s servers that are pre-rendered in a browser component.

# Team Salsa’s Solution

Our team proposed the solution of mirroring the iOS application’s functionality and user interface in a natively developed Android application. It will be coded using the Java programming language, and will use Google’s Android Studio SDK for development. We would create a “container” for the web view content, and engineer software that pulled from Tanga’s JSON data source to display item information. Our team intended to mirror the iOS user interface as much as possible; this includes menus to allow users to access all of the same pages and sections as the iOS app allows. Users are able to choose an item from any sections, view all related item information, add items to a shopping cart, and check out via Tanga’s servers.

# The Team

Team Salsa was formed in late August of 2015 from five students in Joe Clark’s CIS Capstone course. This team was formed to complete the Computer Information Systems capstone project, one of the final challenges for obtaining a Bachelor’s of Science in Computer Information Systems from Arizona State University’s renowned W.P. Carey School of Business. Each team member has brought their invaluable skills to the table, both from their common educational background and varied work histories.

Manoj Panikkar has always enjoyed challenging himself intellectually. This fact becomes very obvious when reviewing his educational history. While double majoring in Accounting and Computer Information Systems, he decided that these courses of study were not enough to quench his insatiable thirst for knowledge and turned his double major into a triple major by adding a Bachelors in Business Data Analytics to his academic curriculum. Even this intense workload did not stop Manoj from being a permanent fixture on the Dean’s list for his entire tenure in Barrett, the Honors College, at Arizona State University. Many facets of Manoj’s diverse skillset were utilized during the capstone project, specifically his knowledge of Java programming and android application development as Manoj spearheaded building the web client that became the team’s first product release.

Apart from pure academic and technical work, Manoj also enjoys playing chess competitively and is a USCF ranked player. Manoj also helps spread his love of learning by working with struggling elementary students through the Arizona Mentor Society. Through his mentorship Manoj has helped many students not only improve their grades but gain a passion for learning as well. His passion for helping others is not just limited to students in elementary school; he also serves as a mentor to incoming freshmen at Arizona State University by sharing his knowledge of how to plan for and build a successful career with them. After graduation, Manoj will be joining KPMG’s Silicon Valley office as a merger and acquisitions specialist. With his broad range of skills, teaching experience, and willingness to help others Manoj is a valuable member of Team Salsa.

Kevin Windhorst is a man defined by his dedication and strong work ethic. He has been able to apply much of his skillset and past experiences while dealing with the many challenges that has faced Team Salsa in the Capstone project. He transferred to Arizona State University in the fall of 2014 after completing much of his lower division coursework at Mesa Community College. Kevin’s work experience has provided him with a unique mix of technical and customer service skills. Kevin has been able to refine his people skills by providing service to thousands of customers during his career. These skills were put to good use by the capstone project as Kevin handled much of the communication with the client (Tanga), his experience showing as he both kept the team informed of Tanga’s requirements and kept Tanga informed of the project's progress through its development.

Kevin was successfully able to gather requirements from the client and communicate those requirements to the development team, a skill honed during his work as a consultant with Deloitte during the summer of 2015. Even though handling communications was his primary role, Kevin was also able to call upon his knowledge of coding gained from his coursework while pursuing a degree in Computer Information Systems at Arizona State University to assist the development team when needed. Once completing his degree, Kevin will be working as an Advisory Consultant at Deloitte’s Phoenix office. Kevin’s extensive experience with customer service and strong soft skills was an invaluable component to Team Salsa’s success.

Mario Padilla is a technically inclined person; having studied subjects related to both computer science and information systems at the Ira Fulton School of Engineering and the W. P. Carey School of Business. Mario’s skill in Java coding and android application development made him ideal to be one of the lead coders for Team Salsa. Mario possesses a great eye for detail which is a great asset not only in his coding endeavors but also assisted him in obtaining a Bachelor’s of Science in both Accounting and Computer Information Systems from the W.P. Carey School of Business at Arizona State University. While personally having strong coding skills, Mario also likes to help others learn coding by teaching introductory coding and Android app development through his volunteering work at goIT App Development boot camp.

Mario strives to be a fixture on campus applying his teaching experience to his current work as the official class tutor for an e-commerce and web development class, while also having maintained his position on the Dean’s list for his entire tenure at Arizona State University. After graduation, Mario plans to pursue a career in software development or enter graduate school for accountancy. Mario’s technical expertise has been a critical component to the successful development of the Tanga android application. His prior experience along with his ability to help and teach others made him a very valuable member to Team Salsa.

Valina Gewargis possesses a broad and effective skillset with work experiences that challenged her interpersonal and technical skillsets alike. This combination helped her to quickly integrate with and become a productive member of Team Salsa. Valina has obtained her strong communication and interpersonal skills through having worked with over a hundred clients at multiple companies. While gaining these skills through work she was simultaneously obtaining and refining new skills by pursuing a Bachelor's in Computer Information Systems at Arizona State University's W.P. Carey School of Business. Starting in May of 2015 she has been interning at APS, the largest utility provider in the state of Arizona.

In her position as an intranet and web developer at APS, Valina has continued to refine skills that she obtained in the classroom. Her experience with professional communication and web development has been utilized in the capstone project though deconstructing the functionality of Tanga’s web application. This gave the team necessary insights into the technical functionality that needed to be recreated by the team, along with expertise on how to effectively integrate functionality provided by Tanga’s web back-end into the new Android Application. Valina’s experience and contributions have been a key component to Team Salsa’s success. Upon graduation, she plans on continuing her work with APS in Phoenix, Arizona.

John Read has held an interest in technology since building his first computer in 2007. He developed an interest in business after starting an Automobile detailing company in 2010. A large set of knowledge from his previous business and technological endeavors, such as assisting with the implementation of a new ERP system at a small CPA firm, has led John to be a valuable contributor to Team Salsa. John decided to switch his major from Finance to Computer Information Systems at the W.P. Carey School of Business at Arizona State University as a sophomore, after rediscovering his passion for business and technology though the aforementioned ERP implementation experience.

His current focus on ERP systems and business databases unfortunately did not have a direct correlation with the web and Android development challenges faced by Team Salsa’s android application project. Even so, he was able to assist the team in various supportive roles by drawing upon his broad high-level knowledge of various technologies. In the professional world, John has worked with business intelligence systems and various reporting tools. Currently he is calling upon his past experience in business intelligence, and ERP implementation with his work at Avnet in Phoenix, Arizona. There he is assisting in report development and implementation of various business intelligence modules for Avnet’s new ERP system. After graduation, John will be working as a business intelligence analyst in Richardson, Texas with State Farm.

With each team member bringing their unique abilities and perspectives to the table, this team of five students was able to combine their wide range of knowledge, skills, and experiences to successfully complete the W. P. Carey’s Computer Information Systems Capstone project. The final product resulted in a functional Android application that recreated the key functionalities of Tanga’s professionally developed iOS ecommerce application. Providing customers with a quick and secure way to take advantage of Tanga’s many deals conveniently through any of their Android devices.

# Functionality of Application (Intended) - iOS

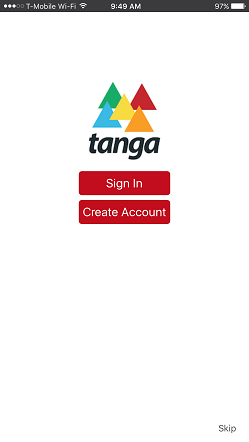
The intended functionality is modeled primarily based on the iOS application, along with initial requirements from Tanga. The initial requirements from Tanga mainly consisted of creating a container to display the navigation items and content on the main screen including the menu bar, product descriptions, pictures, and details, as well as rendering the checkout pages and product pages in the mobile web view. In order to complete the project successfully, the team needed to learn Java, be familiar with basic computer science and mobile development concepts, and understand how to use APIs.

The first task associated with the project was to replicate the application icon from the iOS application to the Android application (Figure 1). In order to do this, Tanga provided Team Salsa with the appropriate image files used for their project, and the resulting application icon for Android would need to be created when users download or view the application in their application drawer.



*Figure 1. Application Icon*

Upon initially opening the application, the iOS version suggests that the user either signs in to an existing account or creates a new user account to access the Tanga application and have their shopping and checkout history saved (Figure 2). An option to skip either signing in or creating an account is also available for users in the lower right-hand corner of the app. However, the sign-in function was not included in the list of initial requirements by Tanga for the first Android version of the application, and was therefore excluded from initial intended functionality when developing the application. Further development beyond the initial Android version may lead to a sign-in page identical to the iOS application.



*Figure 2. Initial Sign-in Page*

The main content page had a mobile list view of Tanga’s products, including the product’s most significant details such as price, original MSRP, and product images (Figure 3). This was achieved by pulling data through JSON generated by the website, as well as using the REST API to gather and parse information in the iOS version, and would need to be done in a similar approach in the Android version. In addition, the main content page has categories at the top of the application—such as Daily Deals, Men, Women, etc.—that can be selected by the user to view different products. Although it is not a feature in the iOS application, the team decided to allow the user to also swipe across tabs to view different categories as an intended function of the new Android application.



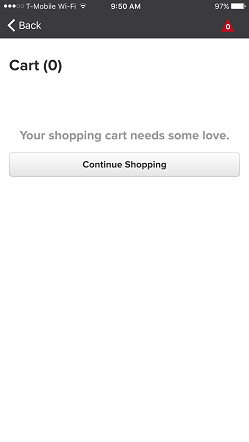
*Figure 3. Main Content Page*

After selecting a product from the main content page, the iOS application also allows the user to see further details about the product and add it to the cart (Figure 4). The user can change the quantity purchased, with the default quantity set to 1, as well as view shipping information, additional product details, and other similar purchases made by users on this page. This function should be replicated on the Android version of the application by using a mobile web view that links directly through Tanga’s website. Parsing the information from the JSON collected on Tanga’s main content page should retrieve the link.

**

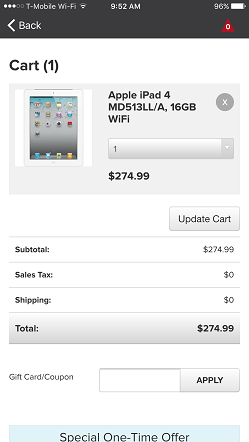
*Figure 4. Add to Cart*

A shopping cart was also added to the iOS version of the application, and was required for the Android version of the application (Figure 5). The cart allows the user to view what products were “checked out” for future purchase, and allows the user to quickly remove products from the cart, or edit the quantity of products being purchased. This function is separate from the checkout process, and can be accessed at any time while the user is either browsing products or in the middle of purchasing a product. This function was not intended for the initial release of the Android application, but may be developed further based on Tanga’s own discretion on application development.

**

*Figure 5. Shopping Cart*

Finally, users can checkout and pay for the product(s) that they have in their shopping cart (Figure 6). This is also an intended feature of the Android application. All cart and credit card information will be processed by Tanga’s website instead of using an internal checkout system on the mobile application itself. This is also achieved through the web view in the iOS application, and will be approached in a similar fashion by using the web view when creating the Android application.

**

*Figure 6. Checkout Page*

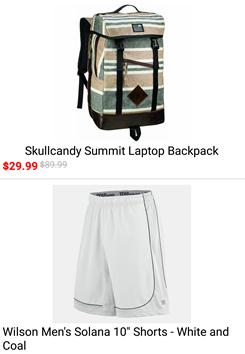
# Functionality of Application (Actual) - Android

The navigation slider allows users to scroll right or left and then taps the category they wish to select (Figure 7).



*Figure 7. Navigation Slider*

The product pane list view gives a photo of a product along with its price and name. Tapping on the product preview pane opens up the full product page (Figure 8).



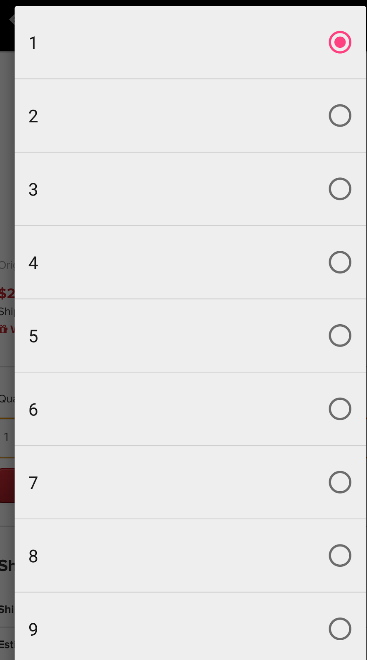
*Figure 8. Product list view*

The product photo carousel allows users to swipe left and right to view different photos of the product (Figure 9). The grey circle at the bottom indicates the where in the carousel the photo that is being viewed is located, while the grey dots indicate other product photos, which the user can swipe to view.

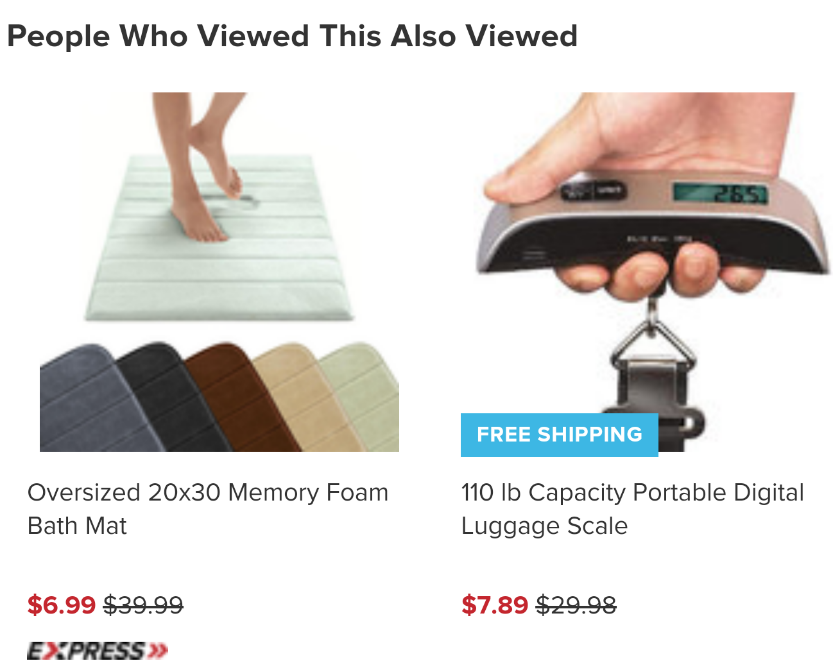
*Figure 9. Product Photo Carousel*

A drop-down menu allows users to select different versions of the item (e.g. item color) and select the quantity of the item they wish to add to their cart (Figure 10). Tapping the “Add to Cart” button automatically brings the user to their Cart with the item added to it.



*Figure 10. Add to Cart Options*

At the bottom of the product page two product preview panes are shown (Figure 11). The products shown here are based off of what other customers have viewed who also viewed the product on the current page.



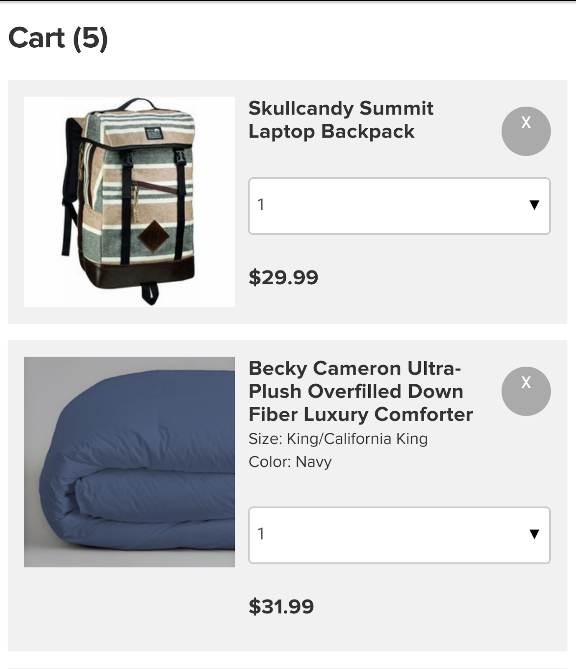
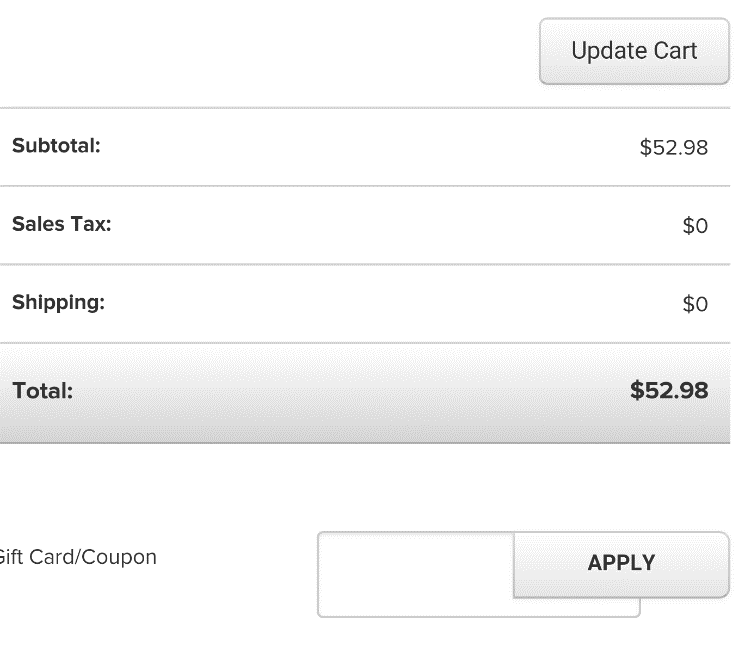
*Figure 11. Similar Product View Pane*

The back button is located at the upper-left corner of the screen and navigates to the previously viewed page (Figure 12).



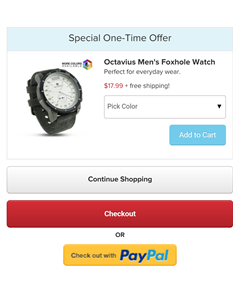
*Figure 12. Back Button*

The shopping cart shows the product preview panes for those products that the customer has added to their cart (Figure 13). The customer can also remove products and change their quantity directly from the cart. An area for the user to enter coupon codes is also present. The cart will automatically calculate the total price of all items selected items minus any discounts, then add in sales and shipping costs, displaying these as separate line items.

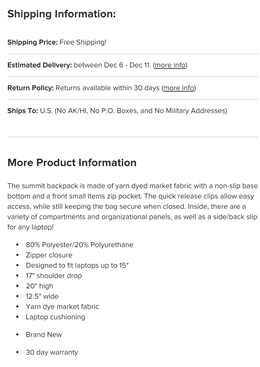
*Figure 13. Shopping Cart*

The checkout button, located at the bottom of the Cart Page, allows customers to choose whether to use a credit card or their PayPal account to pay for items (Figure 14). Tapping either button will prompt the customer to login or create an account. Also, a special deal will be offered to the customer at the bottom of the cart page just above the checkout button.



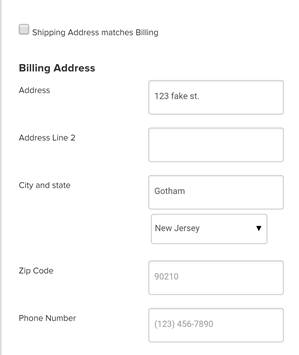
*Figure 14. Checkout Page*

The product page shows more information about the product compared to the product preview pane (Figure 15). This page involves a quick product description at the top along with detailed product info and shipping details below.

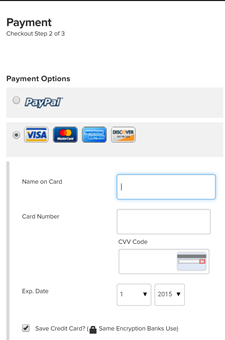
*Figure 15. Product Info Page*

The customer information page has users enter their home address, along with name and phone number when paying for a product (Figure 16).



*Figure 16. Customer Information*

The billing information page has users enter their credit card information or select PayPal to handle their payment (Figure 17).



*Figure 17. Billing Information*

The Tanga application logo is an icon that sits in the app drawer or on one of phone’s homepages (Figure 18). Users can access the Tanga app by tapping it.



*Figure 18. Application Icon*

# Business Issues: Communication

There were some barriers in communication from the start of the project progression. At the beginning of the semester, there were several delays in meeting with a software developer from Tanga to walk our developers through the code and fully understand each segment. After a month, we were able to meet with a developer for code review purposes. At first, Tanga provided an older version of the iOS app developed in Objective-C. Our team discovered later on that there was a newer package coded in Swift. We were not provided a strict set of requirements for the native Android application until several months into the semester. It required strong initiative from Team Salsa to acquire feedback for the web view application developed as a prototype. After obtaining a solid understanding of the needed deliverable, our team had difficulty understanding how Tanga utilized JSON to pull in item data from their servers. It was necessary for our team to contact several Tanga members to find the correct employee to walk us through that part of the application.

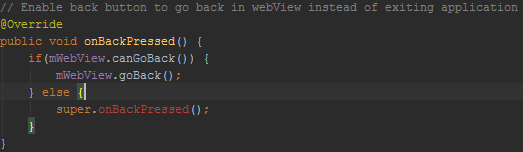
# Technology Issues: Browser-Based Application

Challenges of building the browser-based version primarily consisted of using the built-in web view functionality of Android to display Tanga’s URL in a browser. Android web view uses the phone’s native browser to pull up information on tanga.com, without having browser widgets, a navigation URL, and ignoring web page errors. To get the web view working correctly, we needed to instantiate the WebView class in the onCreate() method, which immediately starts when the application is run. Unfortunately, web view also disables JavaScript by default, which Team Salsa had to explicitly enable when developing the application (Figure 19).

https://lh6.googleusercontent.com/pQjSOCc2GFdXPyBkOQtVMHNPD4W5BJkR2bb8FpyXaBDO7_EprM5VjgVjLYoFcCTKjWD3SbnFP3Swhcj37ZWepjxjDqP2i8ROO63qxd6kmcbgcR7SsVof5IJA9VG3VwhQ9fKWm3Dr

*Figure 19. Enabling JavaScript in Android web view*

After testing the first iteration of the browser version, the main issue and the primary feedback we received with the application was that Android’s built-in back button would close the application instead of taking the user to the last web page that they viewed. To fix this issue, we created a new onBackPressed() method to let the back button interact with the web view instead of closing the application (Figure 20). Finally, the Android application failed to pull in data since it needed to access the Internet for the web view. In order to solve this issue, Team Salsa needed to add Internet permissions to the Android Manifest file.



*Figure 20. Enabling back button functionality in the web view*

# Technology Issues: Native Application

Team Tanga’s main challenge going into this project was developing the native Android mobile application using Java when none of the team members had any previous experience in Java or mobile development. To tackle this challenge, the team members had to familiarize themselves with the programming language and how it is utilized for mobile development. Two of the team members, Manoj and Mario, enrolled in the CIS class Business Systems Development using Java to further broaden their understanding and experience with the programming language. Team member Valina took a different approach by utilizing the book “Java Programing for Android Developers for Dummies” for self-paced learning. Other team members, Kevin and John, applied their past experience and knowledge with other programming languages (C#) to contribute to the project. With the basic understanding of the programming language and programing method, Team Tanga was ready to tackle the development head on.

During the first phase of development, the team was transfixed on replicating the already established iOS application. The main issue was that iOS and Android development used two different programing languages that were very different. The team was unsuccessful in trying to duplicate the code in a different language. The main contributing factors included drivers, development tools, Android studio and Xcode, different project structure, as well as varying programing language classes. The team had to take a step back and reevaluate the project from a different perspective. Instead of just porting the code over to Java, the team decided to focus on the basic features and functions of the app and start by building those features. To help with gathering this information, Tanga’s mobile developer Chris Young, who worked on developing the iOS version of the app, walked Team Tanga through the Swift code for the application and how each part correlated to a specific function of the app. Through this virtual meeting, the team was able to pick out what the main parts of the code intended to accomplish.

When broken down, the main parts of the original swift code connected to the official Tanga site to pull the JSON data, parse through this data, and display the items in a list view. Other functionality included connecting to the web view of the product page and connecting to the cart on the website for checkout. Knowing the features that needed to be replicated was much easier than trying to literally port over the code itself. Moving forward, the team was able to use their understanding of Java to mimic these features in the Android app.

# Technology Stack

The primary program used to develop the mobile application was Android Studio. Android Studio was used to develop as well as test the mobile application during development. Android Studio contains Gradle build support, Lint tools, Pro-Guard app-signing tools, and wizards to create templates for components in common applications. The Volley API was also used to parse and process JSON information extracted from Tanga’s website to display in the application.

# Testing Procedures

Testing is a concept that is extremely important to our team. We understand that ultimately, the end user holds the most important opinion in the process of creating good software. Our goal was to create a functional and intuitive application, but we didn’t want to answer a question that no user was asking. Therefore, we implemented strategic testing in two phases.

The first testing phase was done early on in development. We gathered two separate pools of independent users. The first pool was made up of six randomly selected iOS users. Our intention was to obtain user feedback from Tanga’s fully working application so we could have a competent understanding of the average end user’s experience. Since this testing was done early in the development life cycle, our team determined that user feedback could be helpful in improving the Android app over the iOS version. At this point in time, our team had created a web-view prototype for Android devices. None of the functionality was native, but instead mirrored the mobile Tanga site. Since the mobile site and iOS apps are similar, much of the functionality was duplicated. The second testing pool was made up of two randomly selected Android users. These users were provided the web view as testing software. After the Native Android app was complete, we chose 8 random individuals to comprise the third testing pool. For all three groups of users, we instructed the individuals to use the software for about five minutes, and explore all functionality of the app. We asked them to go through all steps of making a purchase without actually completing the transaction. Each user was then provided the same survey questions to fill out.

These are the five questions provided to each user:

1. On a scale of 1 to 10, (1 being extremely difficult, 10 being extremely easy) how easy to use was this application?
2. On a scale of 1 to 10, (1 being extremely irrelevant, 10 being extremely relevant) how relevant was the product information provided?
3. Did you experience any performance slowdowns/lag when using this software? If so, please explain:
4. On a scale of 1 to 10, (1 being competes poorly, 10 being competes very well), how well does this application experience compare with competing e commerce sites like Amazon or eBay?
5. On a scale of 1 to 10, (1 being not recommend at all, 10 being absolutely recommend), how likely would you be to recommend this application to others?

# Testing Data Analysis

For analysis purposes, our team calculated two average values for each question; one from the iOS users, the other from the Android testers. This information is displayed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Question Number** | **iOS Average** | **Android WebView Average** | **Android Native Average** |
| Question #1 | 9.2 | 5.5 | 8.2 |
| Question #2 | 8.3 | 8 | 8.2 |
| Question #3 | NA | NA | NA |
| Question #4 | 7.3 | 7.5 | 6.5 |
| Question #5 | 6.5 | 6.6 | 8 |

First and most interesting is the disparity between the iOS and WebView average values for Question #1. The low value demonstrates that the WebView is not intuitive or user friendly. This high iOS number dictates that its user interface is strong; clearly it should be mirrored on the Android app as much as possible. When the Native Android app was tested, it saw a strong value of 8.2. All three values for Question #2 were similarly strong. This demonstrates that Tanga is providing useful product information, although it could be refined slightly. For Question #3 there are no calculated values, but users stated there were no significant performance issues throughout the software. Several users did mention that the app should load more than 5 items at a time, which appeared slow but worked as intended. Another observation was that at certain times images took too long to load. It is difficult to determine whether this is a software or network issue, and how practically it can be compensated for. One potential solution would be preloading images in cache or RAM for products the user has not opened yet. The values for Question #4 were average, and almost identical across the iOS and Android WebView apps. The Android Native app scored slightly lower at 6.5, but reviewing the comments helps to understand why. Multiple users claimed that while Tanga had better deals than competing sites like eBay and Amazon, those services had more robust functionality and a wider product catalog. Testers didn’t have an issue using Tanga, but didn’t appear convinced that they would use this site over existing alternatives. This brings up the question: how much can the appeal of this software be improved by developer effort versus Tanga implementing concepts that demonstrate greater value to the consumer? Our data for Question #5 from the iOS and Android WebView users was low average and almost identical. Interestingly, the Android Native app had a higher value of 8. We can conclude that all factors being considered, users are somewhat excited about using this software, but not immensely so. While it is functional and intuitive, testers still see strong value in other existing ecommerce platforms. Several testers stated that the large amount of product information was a significant benefit. Criticisms included a lack of user reviews for products, and no search functionality in the Android app.

# Next Steps

A majority of the initial requested features for the application as well as original requirements have been met and are integrated into our Android application. The next step is to hand over the application, along with the source code, to Tanga’s main mobile developers. Since the application is fully functioning and ready for consumer use, Tanga has a few options in moving forward. The application as it stands now can be published to the Google Play store (Android’s mobile app store) and released to the public. The process is simple and straightforward; it includes registering for a Google Play publishing account, paying a one-time registration fee, and submitting the application for review. Google’s review process is relatively hassle-free and within a few hours the application will be available within the app store for download. Tanga also has the option of releasing the application as a full version (version 1) or as a beta. Typically “beta” implies that the application is still undergoing further development and/or testing for improved functionality.

Tanga also has the option of holding off on releasing the application to the Google Play store and focus on further developing the app with added functions and features. One feature that the Android version of the application has not yet implemented is the “search bar”. Currently, the application does not support search capability. Having a search bar was an original requirement, but due to feature prioritization, it was moved down the list as a lower priority feature. The search bar can be incorporated into the app by using the same JSON data taken from the website to parse, find, and display relevant products.

Another enhancement that was not included as a requirement for our project, but was expressed by Tanga as a future add-on, is the integration of setting up and signing into a customer account through the native mobile app. Customers are currently able to create an account through Tanga’s website where they can access items saved in their virtual cart, as well as saved shipping information for a faster checkout experience. Tanga hopes to add this feature to the mobile application to enhance the mobile shopping experience for their customers. The mobile developers will use the source code we have already developed as the base for these and all other future enhancements.