**UNIVERSITY OF WATERLOO**
**Faculty of Engineering**

# The Development of a Container of Collections of Items for Android

Zazzle Inc.
1800 Seaport Blvd, Redwood City, CA 94063, United States

Lariesa Janecka
Software Engineering 3B
lajaneck
20460089
May 8th, 2016

Lariesa Janecka
271 Westcourt Place
Waterloo, N2L2R8

May 9, 2016

Andrew Morton, Director
Software Engineering Program
University of Waterloo
200 University Ave. West
Waterloo, Ontario. N2L 3G1

Dear Andrew Morton:

This report, entitled “The Development of a Container of Collections of Items for Android”, was prepared during my 3B work term at Zazzle Inc., where I worked in the Software Engineering department.

During my work term at Zazzle Inc. I worked as a member of the Mobile Development Team. I worked primarily in implementing new features for their iOS application and bringing their android application up to feature parody. Every feature that I worked on centred around better displaying product information for a user to purchase.

This report aims to explain and justify the design choices that were made during the development of a method to display collections of items for a android application for the sale of consumer goods.

Acknowledgements and thanks go to Kevin James Carruthers for supplying a LaTeXtemplate to properly format this report.

I hereby confirm that I have received no help, other than what is mentioned above, in writing this report. I also confirm this report has not been previously submitted for academic credit at this or any other academic institution.

Yours,

Lariesa Janecka, 20460089

# Executive Summary

Zazzle's mobile platforms have been lagging behind their web platform for many years. A major functionality that was missing from the android mobile platform was the integration of a user's ability to sort products into easily defined sets. This allows users to manage their own data and categorize things to their liking. It is a large portion of user interaction on the web platform and sorely missing from the android platform.

The purpose of this report is to outline and justify the development of a feature to display multiple collections of products into Zazzle's android [1] mobile application. All design decisions will be explained and evaluated. The qualitative nature of user interface design makes quantitative evaluation difficult and usually obtuse. This report will work to break this problem into three smaller problems that can be evaluated using multi-criteria decision analysis to evaluate the feature's ability to solve each of the sub-problems.

The report looks at the developed feature's ability to accurately display data pertaining to user defined sets of products, allow users to manipulate that data, and persist any changes to the database to ensure data continuity between platforms.

Using this method of decision analysis this report concludes that the feature satisfactorily fulfils all of the desired criteria and provides a solid base for future development. It recommends that in the future a few small bugs that negatively impacted user interaction with the feature be solved and more tests be added to prevent further bugs from going by unnoticed.

# Table of Contents

# List of Tables

# 1 Introduction

One aspect of Zazzle's product is the integration of collections of products. Instead of a single list of liked or favourited items, as many similar shopping apps use, Zazzle allows the user to create multiple different collections of items so that the user can sort products in a more customizable manner. To do this a new user interface and backend were required to allow this new functionality.

The main goal of the feature described in this report is to display collections of products in such a way as to allow the user to sort and manipulate items for future purchase. The most important criterion for a successful solution is for the feature to display collections of products clearly and in an intuitive fashion. There are multiple collections each with their own data and settings, containing multiple products, each with their own data and settings. This data should be displayed in a clean, user friendly way that fits the mobile platform properly.

This criterion will be evaluated based on how intuitive it is as the user's ability to use the feature is paramount. This is a very user facing part of the feature so its visual appearance and ease of use will weigh heavily in the evaluation of a solution. The final method of evaluation is how quickly the user facing representation loads and reacts to changes in the underlying data. Collections can be very large and contain hundreds of products. This large amount of very dynamic information cannot take too long to load before the user gets too frustrated to use the application.

Another criterion to fulfil is to allow the user to manipulate the data associated with collections and their corresponding products. The manipulation of this data includes adding, deleting, and reordering collections and the items they contain. For each collection the user must be able to manipulate its settings including its name and whether it is private or viewable to other users. This criterion will be evaluated based on how well it allows the user to manipulate these values and how intuitive the user interface to do so is, as the manipulation of this information is necessary for the user. As the user makes changes these changes must be reflected in the data visible to the application so that the user knows their changes properly effected the underlying data. For this reason the correctness of the data and available options being displayed is important.

All of the changes to the user interface to allow the new collections format changes to the API are required. These API [2] changes will be the basis of further expansions so the prime criterion for evaluating any solution to this is the ability for it to expand to include more functionality. Solutions will be evaluated on their flexibility, expressed as their ability to be used for multiple purposes. Solutions will also be evaluated on their ability to expand to include more functions. Finally solutions will be evaluated based of their ability provide the required data for the front end in a usable way.

This report is intended for an audience with rudimentary experience using a mobile application. It will describe the problem with more quantifiable criteria and methods of evaluation. It will explore two options for each of the three criteria and evaluate which was the best option. Finally the report will evaluate the implemented solutions on how well they fill the requirements.

# 2 Main Body

## 2.1 Problem Description

The current set up of Zazzle's android mobile application uses many different data structure types to display old data that the website has migrated away from. Products are stored in a wishlist or a likes object that reads older data and has limited functionality. The user interface for these objects is nonintuitive and visually unappealing. This is leading to low user interaction and discourages users from using the application. The android application needs to update is user interface to fit more with material standards implemented in most other android applications, update its interactions to support all functionality that the website does, and update its API layer to use a single object that matches more with the underlying data.

The problem of integrating the ability to properly display collections of products into Zazzle's existing android application has been broken up into three criteria that a viable solution must fulfil. First is to be able to display the relevant information, second is to allow the user to edit the collections' information, and third is to percolate those changes to the back end database.

Due to the qualitative nature of these criteria Multi-Criteria Decision Analysis will be used to determine if the proposed solution is viable. Each criterion will have a set of weighted attributes associated with it. Assigning values to how well a solution fulfils each attribute will provide a numerical grade for that solution. For each of the criteria given in the problem statement a solution must achieve a high enough score to be considered functional.

The first criterion focuses mostly on how the application will present the relevant data. A viable solution must display a users collections so that the user can identify each one at a glance, display the metadata related to a collection and its settings clearly, and display the products in a collection so that a user can easily identify which collection these belong to. These three sub-criteria will be evaluated on their visual appearance, intuitiveness, ease of use, and reactivity.

The weights for each section of this criterion can be seen in Table 2.1. For displaying collections more emphasis is placed on the reactivity of the feature as this is the first element the user sees and has the most data to load. Less weight is given to its ease of use as very little user interaction happens here. For the collection metadata more emphasis is placed on intuitiveness and ease of use since very little is displayed here but there is a lot of user interaction. When displaying the products in a collection very little user interaction is required resulting in less weight being placed on ease of use, but a high understanding of how this data corresponds to other aspects of the feature is required resulting in high scores for other aspects.

| Name | Appearance | Intuitiveness | Ease of Use | Reactivity | Total |
|---|---|---|---|---|---|
| **Collections** | 0.10 | 0.10 | 0.05 | 0.15 | 0.40 |
| **Metadata** | 0.05 | 0.10 | 0.10 | 0.00 | 0.25 |
| **Products** | 0.10 | 0.10 | 0.05 | 0.10 | 0.35 |

Table 2.1: Weights associated with criterion 1

The second criterion focuses on how the user manipulates the data presented in the feature. They need to easily add, reorder, and remove collections, edit their metadata, and add, reorder, and remove products within a collection. The solution will be evaluated for each of these three classes of manipulations based on its functionality, intuitiveness, and correctness. Functionality is how easily the user can find the controls, intuitiveness is how readily the user understands how the controls work, and correctness represents how accurately the controls reflect what is happening.

The weights of each section of this criteria can be found in Table 2.2. When displaying the set of collections it is most important to ensure that the data is correct, if a collection that the user is attempting to edit is missing the entire feature is rendered useless. These controls should be intuitive to the user, but since they will be rarely used it is less important for them to be easy to find. The exact same manipulations are required when displaying products within a collection so all weights are the same. The metadata of a collection will be the most often manipulated as it is here that most data pertaining to a collection is stored so it is most important that these controls are easy to find and understand. All of the functionally required to manipulate the set of collections must also be provided to manipulating the products within a collection, so its weights are the same.

| Name | Functionality | Intuitiveness | Correctness | Total |
|---|---|---|---|---|
| **Collections** | 0.05 | 0.10 | 0.15 | 0.30 |
| **Metadata** | 0.15 | 0.15 | 0.10 | 0.40 |
| **Products** | 0.05 | 0.10 | 0.15 | 0.30 |

Table 2.2: Weights associated with criterion 2

The third criterion is focuses on keeping the data in the database matching the data displayed.

All changes that the user makes must be communicated to the database so that it can update its data accordingly and all data displayed to the user must be properly retrieved from the database. To do this a functional API layer must be constructed. This will be evaluated on its ability to adapt to new contexts, known as flexibility, its ability to incorporate new features, known as scalability, and its ability to support the required functions to retrieve data in a form easy to display and propagate changes from the user to the database, known as functionality.

The weights for this criterion can be found in Table 2.3 The most basic requirement for this criterion is for it to properly support the front end that is currently built on it which is why the functionality attribute is heavily weighted. It is very likely that the requirements for this feature will be expanded in the future to incorporate more functionality which is why scalability is ranked next highest. Finally is is important that other parts of the application have the ability to use this new API to bring more uniformity to the product in the future, which is why flexibility is given high weight as well.

| Name | Flexibility | Scalability | Functionality |
|---|---|---|---|
| **Collections** | 0.15 | 0.35 | 0.50 |

Table 2.3: Weights associated with criterion 3

## 2.2 Criterion 1: Displaying Information

The main goal of this feature is to accurately display each of the collections created by a user, each collection's metadata (data pertaining to the collection's name and settings), and the products in that collection. There is a lot of information allocated to the collection itself including a name, cover image, and various settings, and even more information allocated to each product within the collection. It was decided to display all of the collection's in a grid to allow the user to see them all and scroll through them. When the user selects a collection from that grid they are taken to a page with all information pertaining to only that collection. This page includes a grid of products within that collection similar to the previous page so that the user can view what products they've put in that collection. Selecting a product from that grid will take the user to the already exiting product page.

Each cell in the grid displays basic information about the collection, its name, cover image and how many items are in that collection. The visual design of the collection grid cells follow material design standards [3] to visually align with other android applications. This same grid design is used for displaying the set of products within a collection on the collection page so it also follows material design. The collection metadata is displayed inside a pop up on that appears when the user clicked a button located on the collection page. This pop up displays each setting in a list with a description and appropriate selector. This pop up is visually designed to adhere to material guidelines.

Due to all aspects of this criterion closely adhering to the material guidelines set out by Google for user interfaces, each will score highly in the visual appearance category. The display of the set of collections in a grid displays it in a fashion that most users recognize from many other android applications making it fairly intuitive if not completely obvious. Each cell in the grid links to the collection's page within one click in a clear and intuitive fashion making it very easy to use. The grid format allows the code to only load information for the collections that can be seen by the user and only retrieve more data when the user scrolls the page. This greatly increases the responsiveness of the application as there is less data to load and loading can be done in a fashion that does not effect the user. Some reactivity is lost due to more data having to be fetched when the user looks for it which can effect the scrolling of the page.

Each setting in the collection metadata pop up contains a clear label that describes what the setting is for and an appropriate selector to encourage the user to input correct data for the setting resulting in a high level of intuitiveness. Accessing the settings pop up requires the user to click a labelled button which makes the data slightly less accessible, but once the pop up is shown, changing a setting is very easy due to the carefully chosen selectors. As stated earlier reactivity is not a consideration for this sub-criterion as the settings pop up does not display data that needs to react properly.

The displaying of products within a collection in a grid on the collection page makes it very obvious which collection the product belongs to. The data is closely tied to the collection through its spacial location and using the same design for its cells, resulting in a high level of intuitiveness. A single click on this grid leads to the product page that the user already has experience with, making it very easy to use. The loading of the product grid works exactly the same as the loading of the collections grid with all the same pitfalls so it will receive the same score.

The above described scores are summarized in Table 2.4. When the weights are applied to these scores, criterion 1 receives an overall score of 83.9 which is well within passing margins of over 50.

| Name | Appearance | Intuitiveness | Ease of Use | Reactivity |
|---|---|---|---|---|
| **Collections** | 90 | 95 | 80 | 70 |
| **Metadata** | 90 | 80 | 75 | N/A |
| **Products** | 90 | 95 | 95 | 70 |

Table 2.4: Values associated with criterion 1

## 2.3 Criterion 2: Manipulating Information

The first page the user sees is a grid where each cell represents a collection. The user needs to be able to add, reorder, and delete collections. To add a collection there is a simple button with a plus sign that presents a form of data through which the user can add all relevant information for the new collection. To delete a collection you tap a small ‘x’ in the collection grid cell. This presents a warning asking if the user is sure that they want to delete that collection. To reorder the set of collections the user just has to select and drag the cells within the grid.

All three of these commands follow the current accepted norm for managing data so it should very intuitive to the user as they have experienced this in other applications. Some points are lost in its intuitiveness score because the grid reordering must be triggered by a long click before the user can drag elements. User testing amongst engineers in other departments at Zazzle showed that this was not their first instinct when presented with the grid. It was more common for them to assume that reordering elements was not supported. All three actions are supported, however, so this sub-criterion gets full marks for functionality. The feature would have received full marks for the correctness of its data, but reordering elements was prone to concurrency issues where the user moving an item around too quickly disrupted the calculations for where it should go resulting in it be placed in a different location than what the user intended.

When a user eventually made it to a collection page they gain the ability to manipulate a collection’s settings. This worked very well due to the simplistic nature of the data being displayed. An entry existed in the settings pop up for each value that could be changed resulting in perfect marks for functionality. Nearly all of the labels for each setting accurately described what a setting controlled. During user testing it was found that not all of the selectors chosen for setting accurately represented what the user thought the data should be. This will have to be fixed in the future. The correctness of the data being presented was nearly flawless. This report would like to give this criterion full marks for correctness but a comprehensive test suite has not, as of yet, been run so it is possible that there exists sequences of actions that could result in incorrect data.

The final sub-criterion required the user to be able to add, reorder, and remove products to a collection. This works nearly identically to the way that the user manipulates the set of collections with the exception of adding products. During A/B testing users found that it was much easier to add a product to a collection from the product's page than it was to add a product from the collection's page. This was mainly due to the user having to look up the product when adding it to the collection from the collection page. The more common use case was a user browsing through products and then choosing to add it to a collection. To this end a button was added the product page that presented a list of collections that the user could add the product to. This proved to be far more intuitive to the user resulting in a near perfect intuitiveness score for this sub-criterion. The adding and deleting of products from a collection worked nearly flawlessly, but this still suffered the same problems when reordering products within a collection leading to a reduced correctness score.

The above described scores are summarized in Table 2.5. When the weights are applied to these scores criterion 2 receives an overall score of 84.5 which is well within passing margins of over 50.

| Name | Functionality | Intuitiveness | Correctness |
|---|---|---|---|
| **Collections** | 100 | 65 | 70 |
| **Metadata** | 100 | 85 | 90 |
| 1 **Products** | 100 | 90 | 70 |

Table 2.5: Values associated with criterion 2

## 2.4 Criterion 3: Interacting with the Database

The collections feature described in this report are all built upon a single object called a collection. This object holds all of the data required for criterion 1 to display everything properly. It also contains all of the functions required by criterion 2 to manipulate this data. This object also contains many functions that can be used to send and receive data from the database. This allows changes that the user has made through the mobile application to persist after the application has been closed. It also updates all of the data that the website uses. Since all mediums of Zazzle's product access the same database any data presented will be properly updated to match changes made on the mobile application.

Previous to the collection feature described in this report there were other data structures being used by the mobile application to display similar, if not limited amounts of this data. All of these structures worked in different ways and required constant maintenance. In order to accommodate these older structures functionality was added to the collection data structure to allow it to be converted to and from these older deprecated structures. This was originally put in to make the transition easier, but was left in to allow the functionality of these other structures to be maintained. By including the conversion functions it was proven that the collection data structure could be used in a flexible fashion to encompass many different uses. This resulted in a high flexibility score.

The collection data structure is modelled as a set of data points and functions for manipulating them. This makes the object very modular making future additions much easier. To add more functionality to the collections object all that is required is adding whatever data points are needed and the functions to manipulate them. During the initial development of the collections grid more functionality was required than had initially been planned, particularly to accommodate reordering objects in the grid. This proved to be a nearly trivial process with no problems encountered. This leads to a very high scalability score as there is definitive proof that scaling up the responsibilities taken by the object can be done very easily.

The collection object initially had many responsibilities and it adopted many more as the project grew in scope. Its core functionality surrounded a set of functions that were used by the user interface to present and manipulate grid of collections, present and manipulate a collection's metadata, and present and manipulate a grid of products in a collection. These six functions are provided to the user interface along with corresponding tests to ensure their accuracy. Further functionality and helper functions also exist on the collection data structure to increase its reuse, but its core functionality is perfectly provided with ensured functionality, This can be evaluated as a perfect functionality score.

The above described scores are summarized in Table 2.6. When the weights are applied to these scores criterion 3 receives an overall score of 95.75 which is well within passing margins of over 50.

| **Name** | **Flexibility** | **Scalability** | **Functionality** |
|---|---|---|---|
| **Collections** | 95 | 90 | 100 |

Table 2.6: Values associated with criterion 3

# 3 Conclusions

The scores of each criterion are summarized in Table 3.1. It is easy to see that each one easily passed the required 50 marks easily. All three received well over 80 marks which displays a high level of satisfaction in the feature's ability to fulfil its requirements.

Each of the three criterion outlined in this report are very easily fulfilled by the feature described

| Name | Score |
|---|---|
| Criterion 1: Displaying Information | 83.9 |
| Criterion 2: Manipulating Information | 84.5 |
| Criterion 3: Interacting with the Database | 95.75 |

Table 3.1: Summary of scores

in this report so it is accurate to say that this feature is ready to be incorporated into Zazzle's android mobile application. Not every aspect of this feature received a perfect score which implies that there is still room for improvement. The criteria outlined in this report are considered the bare minimum of what this feature must include to be put into use.

There are many more functions and side features that have yet to be incorporated that were not included in these requirements, but it has been shown that a very solid base with a proven ability to expand has been built for this which should make adding future capabilities easy. The interface as well is a very solid base. It set a precedent for using Google's material design guidelines which should encourage a clean and user friendly interface with all features to be added in the future. The user interactions have been well defined and implemented in ways that a user can easily understand and use. All features that will be added onto this will have a good baseline to follow with how they implement their user interactions.

In summary, this report concludes that the collection feature has soundly fulfilled its required functionality and built a very solid base for the future to expand upon and build more features for.

# 4 Recommendations

There were many bugs still within the feature, a major one being the errors introduced when reordering items in a grid. These all need to be addressed in the future. It is likely that these problems come from using android native constructs not as they were intended. A possible solution is to seek out third party libraries that have built a maintained solution to this. It is also possible that android will update in the future to better support these interactions. It is also recommended that future developers build a comprehensive test suite to support the user facing portions of this project. The backend is very well tested, but many of the bugs mentioned in this report can be easily tested for with a frontend test suite. This is universally accepted as good practice by the industry and as such the project should follow it.

# Acknowledgements

Acknowledgements and thanks go to Kevin James Carruthers for supplying a LaTeXtemplate to properly format this report.

# References

[1] Google. (2015, Oct.) Android documentation. [Online]. Available: http://developer.android.com/guide/index.html

[2] APIStandards. (2015, Oct.) Api standards. [Online]. Available: http://www.api.org/standards/

[3] Google. (2015, Sep.) Material design. [Online]. Available: https://www.google.com/design/spec/material-design/introduction.html#introduction-principles