



**Technovation** iridescent  
**CANADA**

## Unit 2 – Market Research

# Check in: Share App

- What app idea came out of your brainstorm session?
- What did you think about the experience of choosing an app?
- What is your mission



## Moment of Truth

# Market Research

- ➔ Develop a consumer research plan
- ➔ Ask the right questions for competitor analysis
- ➔ Create a market research report for your app

# Why startups fail and succeed

“The vast majority of startups fail NOT because they could not build a great product or technology, but because **no one wanted the product!**”

- *Steve Blank*



## Oakley Thump Sunglasses (2007)

Sunglasses with a built in MP3 player –  
only 256 MB flash memory – cost \$495  
(<http://blog.laptopmag.com/worst-gadget-flops-of-all-time?slide=7>)

- ✦ The sunglasses are unattractive
- ✦ You need a separate MP3 player for when you're not wearing the sunglasses
- ✦ The sunglasses are also out of on cloudy days
- ✦ It doesn't hold that many songs
- ✦ It is expensive
- ✦ Audio controls were difficult to use



## Twitter Peek (2009)

Twitter fans check their feed and post updates –  
only 20-character previews – \$199  
(<http://blog.laptopmag.com/worst-gadget-flops-of-all-time?slide=11>)

- ✦ Only helps people to use one service (Twitter)
- ✦ You can only see 20-character previews when Twitter posts are 140 characters long (more clicks)
- ✦ Expensive for serving only one function
- ✦ There were plenty of free apps for cheaper smartphones that did the same thing

# Knowing you have the right idea



Are you solving a problem that consumers have and will pay for?

Are other companies solving this problem and is your solution better?

# Know Your Customers

- ➔ You'll never be sure until you ASK!
- ➔ Does your target customer experience the problem your product solves?
- ➔ Are customers aware that they have this problem?
- ➔ Would your target customers pay for a solution like yours? How much would they pay?

# Two types of interviews

## Validating the Problem:

- If your app existed, would people use it? How often?
- What currently fills that need?
- Where is there a need for [describe what your app does]?

## Validating the Solution:

- Which features are most important for your app to have?
- How much would people pay for your app?
- Does your app match the customer's lifestyle?

# Customer Interviews: Preparing

## Beware of yourself!

- We are shy
- We are biased
- We fear rejection
- We are lazy
- We don't listen hard enough



## Finding interviewees: Fish where fish swim

- Introductions from family, friends and colleagues
- Online interest groups
- Industry conferences
- Entrance/exit of events, bus stations, coffee shops, etc.
- Classified ads



At first you'll suck at it but that's OK

- Practice with friends/family/colleagues
- Test your questions
- Leave the most promising interviews for last



## Stop when the feedback gets repetitive



- Keep doing interviews until you stop learning something new
- It could take 10 or 50 interviews
- If this doesn't happen your customer segment could be too broad, so narrow it down



# Sample Survey: Angry Birds



**Angry Birds** is a mobile app game

## *Survey Questions:*

The goal of this survey is to find a group of people that can give you information about your idea/ product / market opportunity

1. Do you enjoy puzzle games? (yes or no) \_\_\_\_\_
2. How satisfied are you with your options of mobile phone games? (1= not satisfied, 10 = very satisfied) \_\_\_\_\_
3. How can we contact you to talk further about this? \_\_\_\_\_

## Resulting Metrics (data we received):

Divide the number of people who responded in a selected way by the total number of people who completed the survey.

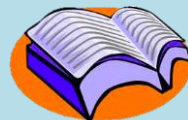
1. 30/50 said yes, they enjoy puzzle games
2. 12/30 reported a satisfaction level of less than 7
3. 40/50 included at least one way to contact them

## What we learned:

- We have a good market size for our app
- Most people are fairly satisfied with their options of mobile games
- People are interested in talking to us about it

# Activity: Create a survey

- **10 min** to list questions
- Good surveys are short
- Pick the top 5-7 questions on your list
- Multiple choice questions with checkboxes are easier for people to answer
- You can allow for more than just one answer
- Also, feel free to just talk to people and take notes. Get some video footage if you can. This may be helpful for your final pitch video.



Create your survey in  
your *workbook*

# Distributing your Survey

Now, get your survey in front of the right people!

- ➔ Use Survey sites such as Survey Monkey - [www.surveymonkey.com](http://www.surveymonkey.com)
  - Distribute the link over social media to your target customers
- ➔ Use Twitter surveys
- ➔ Put your questions on Facebook
- ➔ Email your survey
- ➔ Conduct your surveys in person

**Market Research - Product Template**

1. What's your first reaction to this new product idea?

Very positive

Somewhat positive

Neutral

Somewhat negative

Very negative

2. How well, if at all, does the word "INNOVATIVE" describe this new product?

Extremely well

Very well



# Analyzing your results

- ➔ Compare your findings to your assumptions
- ➔ Questions to consider:
  - What conclusions can you draw?
  - What are some trends you see?
  - Did you get the information you were looking for?
- ➔ Use your survey results to refine your app idea



# Competitive Analysis

- ➔ Competition means other people also believe the problem needs solving – which is good

BUT

- ➔ You need to be able to say why your customers should choose you

# Knowing your competitors

- ➔ Branding & Public Image
  - How well known are they?
  - What is their reputation?
- ➔ Market, Pricing, and Revenue
  - How do they make money?
  - How big are they?
- ➔ Technology
  - What kind of features does the app have?
  - What kind of features is it missing?
  - What feedback do users have?

# Find competitor data

- ➔ Use Google searches to find competitor apps & startups
- ➔ Search Google Play and iOS App Store
- ➔ Use [wikipedia](#) articles about the startup/app to get stats on revenue, years in operation, size of team etc.
- ➔ Study their websites, check out their social media (Twitter, Instagram ...)
- ➔ Look at their apps



# Competitor example

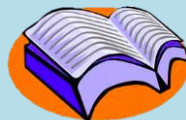


## Apple vs Samsung

- ➔ Branding & Public Image
- ➔ Market, Pricing, and Revenue
- ➔ Technology

# Prepare a Market Research Plan

- ➔ For next week - Ask yourself:
- ➔ Do you feel well prepared and understand the market you are entering?
- ➔ Are you going to make any changes to how you develop your app?
- ➔ What are your strengths and weaknesses compared to your competitors?
- ➔ What will be your selling points against your competitors?  
What sets you apart?



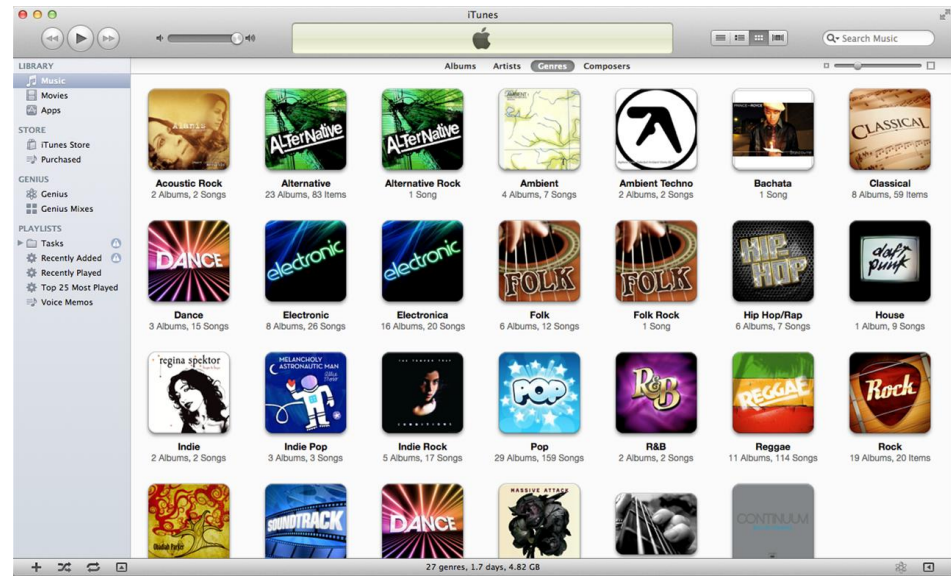
Create your market research plan in your **workbook**

# App Inventor - Databases

- Databases organize the collection and retrieval of data
- They're intended to organize, store, and retrieve large amounts of data easily

## Examples:

- iTunes is a database for all your music
- Google has a database of websites which is used for Google Search
- Facebook is a database of people



# Types of Databases

Remember the [Colored Dots tutorial](#) from last session that introduced databases?

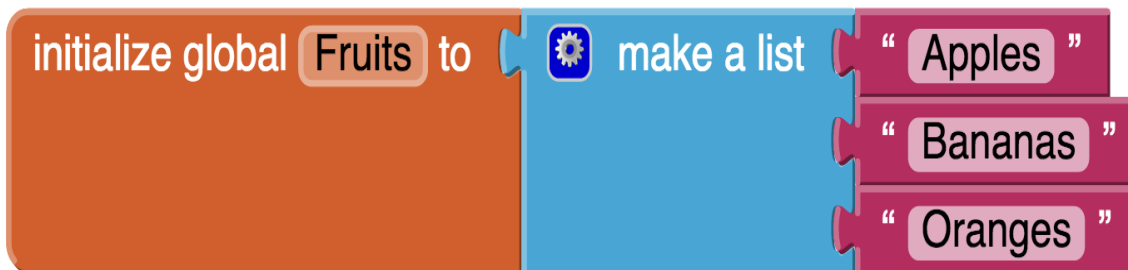
- If you want to save data after the application is closed you ***must*** use a *database*

## **Two types of databases:**

- [TinyDB](#) stores data directly on the phone and will only be used by that phone (hand-set)
- [TinyWebDB](#) stores data on a web database that can be shared among phones

# Organizing data

- ➔ Programmers use lists to organize data in their programs.
- ➔ Lists can hold multiple pieces of data and they're easy to get data from.
- ➔ You may have made a to-do list or a grocery list before, and lists in programming are very similar.



## List Name: Fruits

- Apples (Index = 1)
- Bananas (Index = 2)
- Oranges (Index = 3)

# Using Lists in a Database

Non-visible components



TinyDB1

```
call TinyDB1 .StoreValue
  tag "Food"
  valueToStore get global Fruits
```

```
call TinyDB1 .GetValue
  tag "Food"
  valueIfTagNotThere ""
```

```
call TinyDB1 .StoreValue
  tag "MyAge"
  valueToStore 16
```

```
call TinyDB1 .StoreValue
  tag "Favorites"
  valueToStore make a list
    "Learn how to Code"
    "Visit Family"
    "Listen to Music"
```

# Creating a To-Do list app

1. Use a label to tell your user what your app will be about!
2. Add a textbox for your user to enter their to-do item into
3. Add a button for the user to click once they entered in their to-do item
4. Add ListView to your screen. This is how the user will see their to-do list.
5. Add a save button to your screen for saving the user's to-do list to a database and a reset button to erase the list.
  - We renamed these buttons 'Save' and 'Reset'
6. *Optional:* We added two horizontal arrangements to organize our layout

# Creating a To-Do list app

The screenshot displays the Xamarin Studio IDE for a project named "ToDoList". The interface is divided into several panels:

- Palette:** Lists various UI components under "User Interface". Red arrows point to "Button" (Step 3 & 5), "Label" (Step 1), "ListView" (Step 4), "TextBox" (Step 2), and "Layout" (Step 6).
- Viewer:** Shows a mobile app preview. It includes a title bar "Screen1", a text input field with the placeholder "Enter what you need to do!", an "Enter" button, a "Save" button, and a "Reset" button. The status bar shows "9:48".
- Components:** A tree view showing the hierarchy of components: Screen1, Label1, HorizontalArrangement1, TextBox1, Enter, ListView1, HorizontalArrangement2, Save, and Reset.
- Properties:** Shows the properties for the selected "ListView1" component, including BackgroundColor (Black), ElementsFromStrings, Height, Width, Selection, SelectionColor (Light Gray), ShowFilterBar, TextColor (White), and TextSize (22).

Red arrows and text labels indicate the following steps:

- Step 1: Label
- Step 2: TextBox
- Step 3 & 5: Button
- Step 4: ListView
- Step 6: Layout



# Add your database

The screenshot displays the ToDoList application interface. At the top, there is a green header bar with the title "ToDoList" and navigation buttons: "Screen1", "Add Screen ...", and "Remove Screen". On the right side of the header, there are buttons for "Designer" and "Blocks".

The interface is divided into several panels:

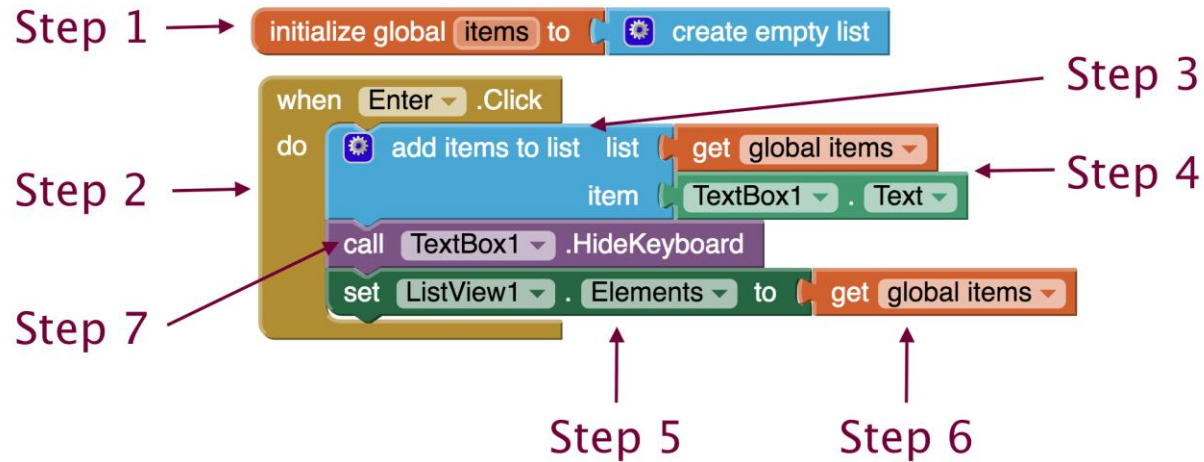
- Palette:** A vertical list of categories and components. The "Storage" category is expanded, showing options: "File", "FusiontablesControl", "TinyDB" (highlighted with a red arrow and the text "Step 7"), and "TinyWebDB".
- Viewer:** A central area showing a preview of the application screen. The screen displays a text input field with the placeholder "Enter what you need to do!", an "Enter" button, and a "Save" button. Below the preview, there is a "Non-visible components" section with a "TinyDB1" component icon.
- Components:** A tree view of the application's components. The "TinyDB1" component is highlighted in green. Other components include "Screen1", "Label1", "HorizontalArrangement1", "TextBox1", "Enter", "ListView1", "HorizontalArrangement2", "Save", and "Reset".
- Properties:** A panel on the right showing the properties for the selected "TinyDB1" component.

At the bottom of the interface, there is a "Media" section with an "Upload File ..." button.

# Managing a list of to-dos

1. Make a variable called *items* that will hold your to-do list data. Start the variable out as an empty list since your user hasn't entered any data yet.
2. Grab a `button.click` event handler for your enter button.
3. Get an "add items to list" block from "Lists". Put it inside your *enter.click* event handler.
4. Put your variable *items* for the list name and "`TextBox1.Text`" as the item to be added to the list.
5. Grab the "set `ListView1.Elements` to" block and put it underneath your "add items to list" block.
6. Add your *items* variable to the "set `ListView1.Elements` to" block.
7. *Optional:* To make our app easier to use we hid our keyboard when the user pressed enter. To do this, click on `TextBox1` and grab the "`TextBox1.HideKeyboard`" block.

# Managing a list of to-dos



# Deleting entries

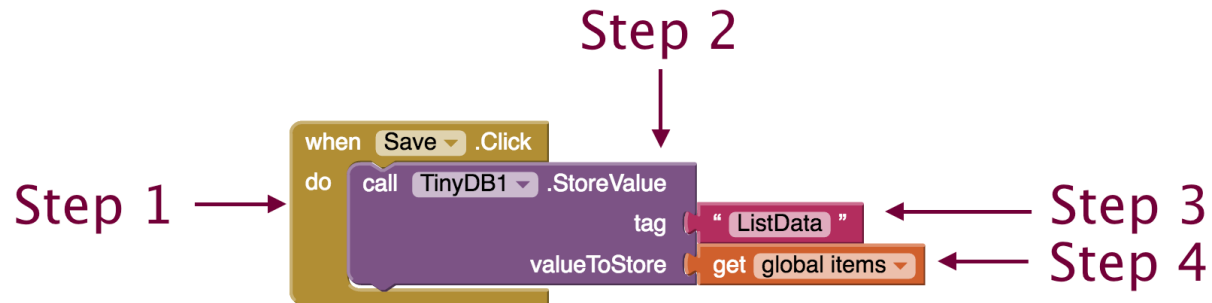
1. Grab the “ListView1.AfterPicking” block.
2. Create a local variable called *index*. You can set it to whatever you like to start out. We set our to zero.
3. Get the “set ‘name’ to” variable block and set it to the variable *index*.
4. Grab the block that says “index is list, thing, list” and attach it.
5. Assign “thing” to be “ListView1.Selection” and the “list” to be the *items* list.
6. Grab the “remove list item block” and put inside your local variable block
7. Set your *items* variable to be the “list” and your *index* variable to be the “index”
8. Grab the “set ListView1.Elements to” block and place it below your orange variable block. Connect your *items* variable to it.

# Deleting entries



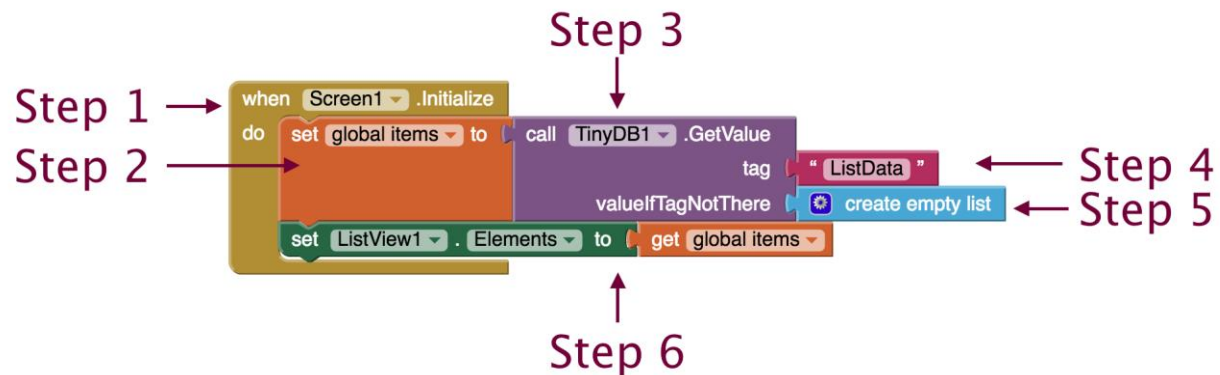
# Saving the list for next time

1. Grab the when “when button.Click” block for your save button
2. Add the “call TinyDB1.StoreValue” block
3. Add a “tag” name
4. Add your *items* variable to “valueToStore”



# Using a saved list

1. Grab the “when Screen1.Initialize” block
2. Grab a “set ‘variable name’ to” block and set the variable to be your *items* list
3. Call the database using “call TinyDB1.GetValue” block
4. Enter the tag name you used to save the list for the “tag”
5. Put the “create empty list” block for “valueIfTagNotThere”
6. Grab the “set ListView1.Elements to” block and attach it to your *items* variable



# Next Steps

- ➔ Work on your market research plan
- ➔ Start your surveys
- ➔ Complete your to-do list app
- ➔ Email or post on Facebook if you need help!

