Logo%20Main%20200

**Boston University**

**Electrical & Computer Engineering**

**EC464 Capstone Senior Design Project**

User's Manual

*Trivia Throw Toy*



Submitted to

Eugene Kolodenker

eugenek@bu.edu

by

Team #24

Team 3T

Team Members

Jacob Dansey [jdansey@bu.edu](mailto:email1@bu.edu)

Changshuo Fu [fcs1994@bu.edu](mailto:email1@bu.edu)

Christine Low [clow@bu.edu](mailto:email1@bu.edu)

Urvashi Mohnani [umohnani@bu.edu](mailto:email1@bu.edu)

Neil Sanghrajka [neils95@bu.edu](mailto:email5@bu.edu)

Submitted: 04/07/2017

#### Trivio

#### Table of Contents

Executive Summary 3

1 Introduction 4

2 System Overview and Installation 5

2.1 Overview block diagram 5

2.2 User interface. 6

2.3 Physical description. 9

2.4 Installation, setup, and support 10

3 Operation of the Project 12

3.1 Operating Mode 1: Normal Operation 12

3.2 Operating Mode 2: Abnormal Operations 14

3.3 Safety Issues 14

4 Technical Background 15

5 Cost Breakdown 18

6 Appendices 20

6.1 Appendix A - Specifications 20

6.2 Appendix B – Team Information 6

# Executive Summary

Trivia Throw Toy

Team 24 – 3T

The Trivia Throw Toy is a ball-shaped toy that speaks in a clear voice a trivia fact that has been intelligently pulled from Internet sources upon detecting it’s been thrown or shaken. The device aims to fill the void in conversation by providing engaging topics of discussion. The user will have the ability to hear facts by category and also up vote/down vote facts to enhance other user’s experience. A mobile application will be used as the interface for picking categories, managing Wi-Fi and voting. An accelerometer is used within the toy to detect motion. Upon detection, a fact is pulled from the proper categories cache and spoken aloud through a speaker. If the cache is near empty, a call to a cloud server is made to pull more facts in.The server handles the pulling in of “fun” facts from various Internet sources. Also, high quality rechargeable battery is used as power supply to provide at least 6 hours useage of the Trivia Throw Toy.

# Introduction

**The problem**

Today’s smartphone allows for a constant connection to the Internet and social media. Although smartphone usage has made life more convenient, it has brought about adverse effects in the realms of mental well-being, interpersonal relationships, and physical health. Today’s adolescents are highly receptive to these new forms of media as they are the first generation growing up surrounded by this technology. It is pertinent that kids are brought away from the screen and into interactions with their peers. Solutions such as not giving them a smartphone, are beneficial but do not take the next step of actively promoting face-to-face engagement. Kids need a fun alternative to their phones in order to engage with their peers on their own accord.

**Our Solution**

There is a clear and present need for a fun and engaging way to promote conversation and learning. The Trivia Throw Toy fills this need by adding a twist to the age-old experience of throwing a ball. It states a fact upon throw, providing educational avenues for conversation, gamified to maintain interest. This encourages people to connect in more meaningful ways and can be useful as an “ice breaker” or a form of entertainment in office and educational environments or simply around the house with friends. Although our product is targeted for a younger crowd, we hope to be applicable for all age ranges 7+. We aim to get people away from the screen and into the real world.

The Trivia Throw Toy is a ball-shaped toy that speaks in a clear voice a trivia fact intelligently parsed from Internet sources upon detecting a throw or shake. Trivia facts can be chosen by category depending on the user’s interest. An accelerometer is used within the toy to detect motion, allowing for varied user experiences ranging from playing catch to simply shaking the device. Upon detection, a fact is pulled from the proper categories and played clearly and audibly through a speaker. A cache of facts is required so that users can bring the toy outdoors and play without internet connection.

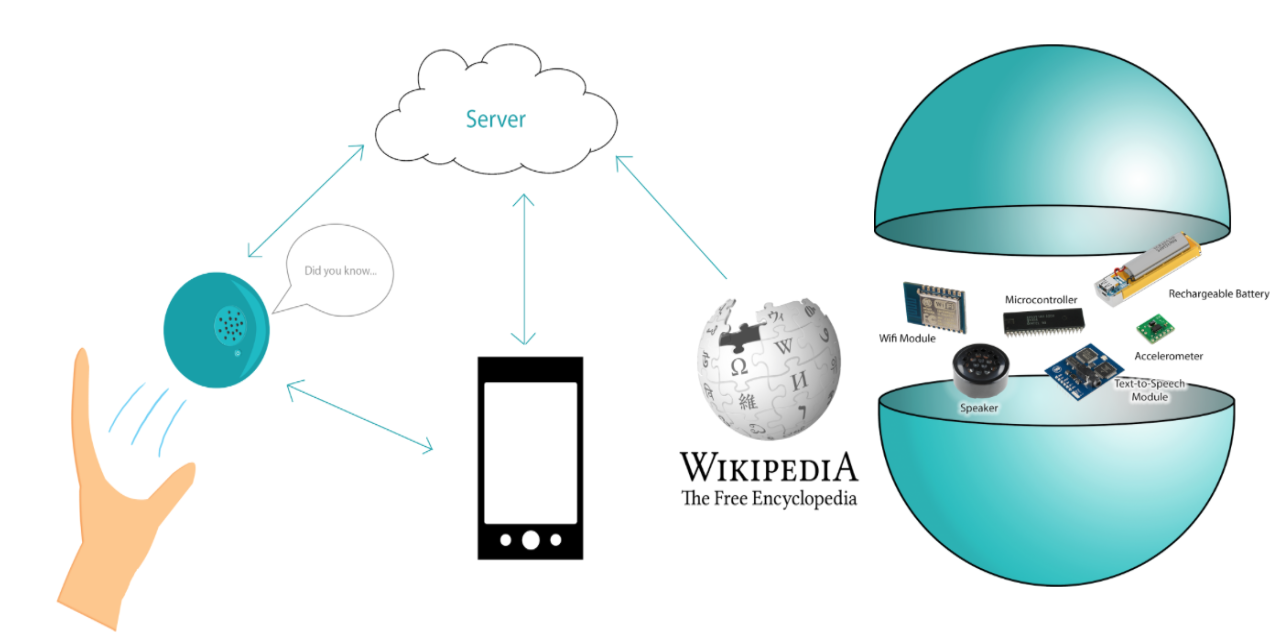
**How are we different?**

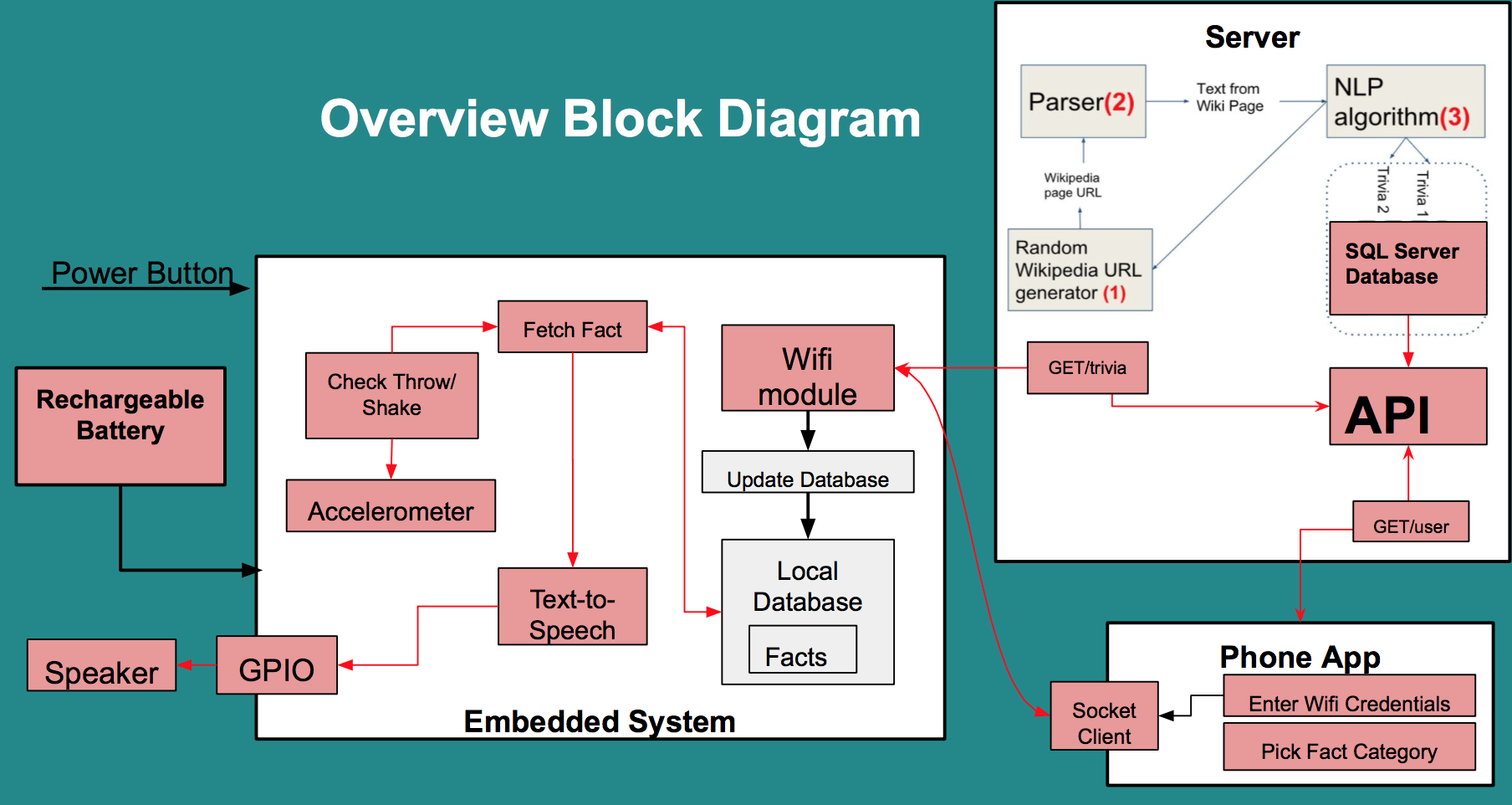
The user will be able to use a mobile application to connect the toy to WiFi. It is also used to customize facts by category and also upvote/downvote facts to enhance other users’ experience. Categories come in a wide variety in order to be applicable to different audiences, such as pop culture or biology. Voting allows the product to improve itself by crowdsourcing feedback on facts. The mobile application also displays a list of all previously heard facts for users to look back on. The server handles the parsing of facts from different fact generating websites, and stores only facts of 140 characters or less. By parsing the Internet for trivia sources, it can maintain a constant flow of new and interesting facts. This differentiates us from our competitors who have a limited shelf life because they use a static source of information. A constantly updating pool of facts gives our product a competitive edge in terms of being highly adaptable. The toy is about the size of a medium beach ball, allowing it to be thrown and catch easily.

# System Overview and Installation

## Overview block diagram

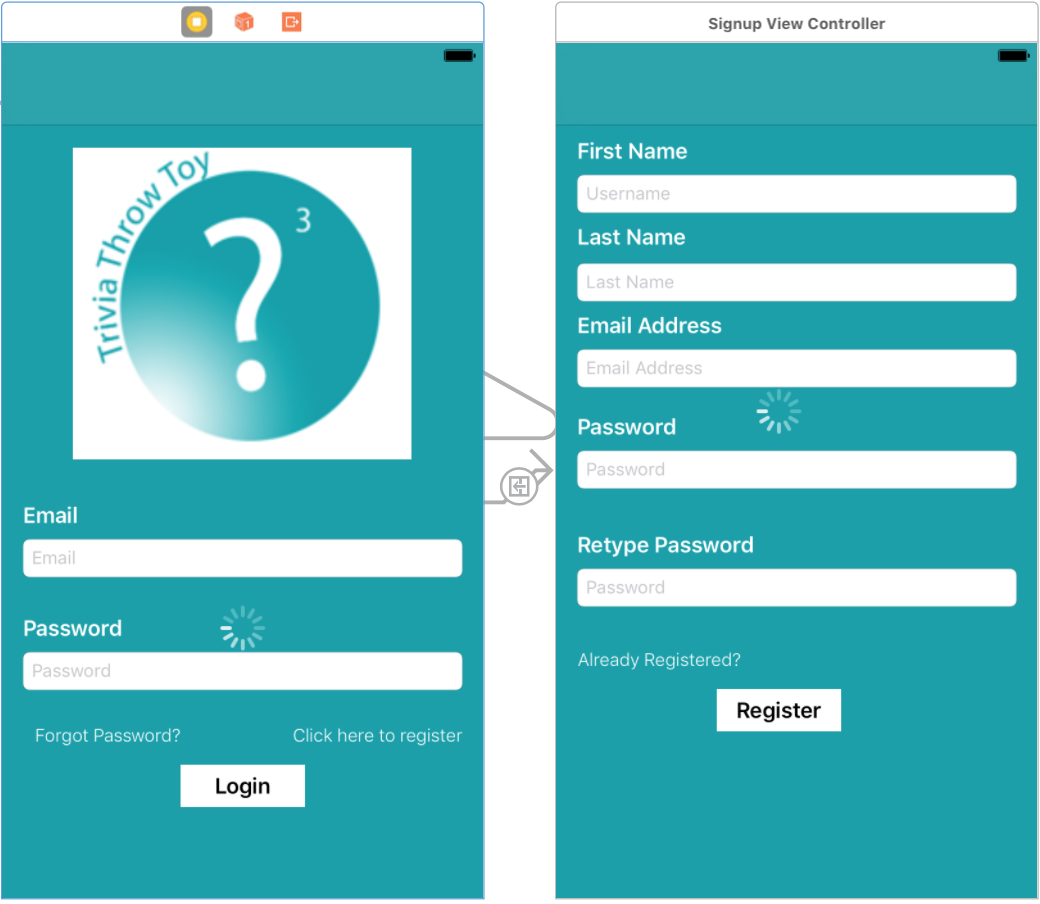
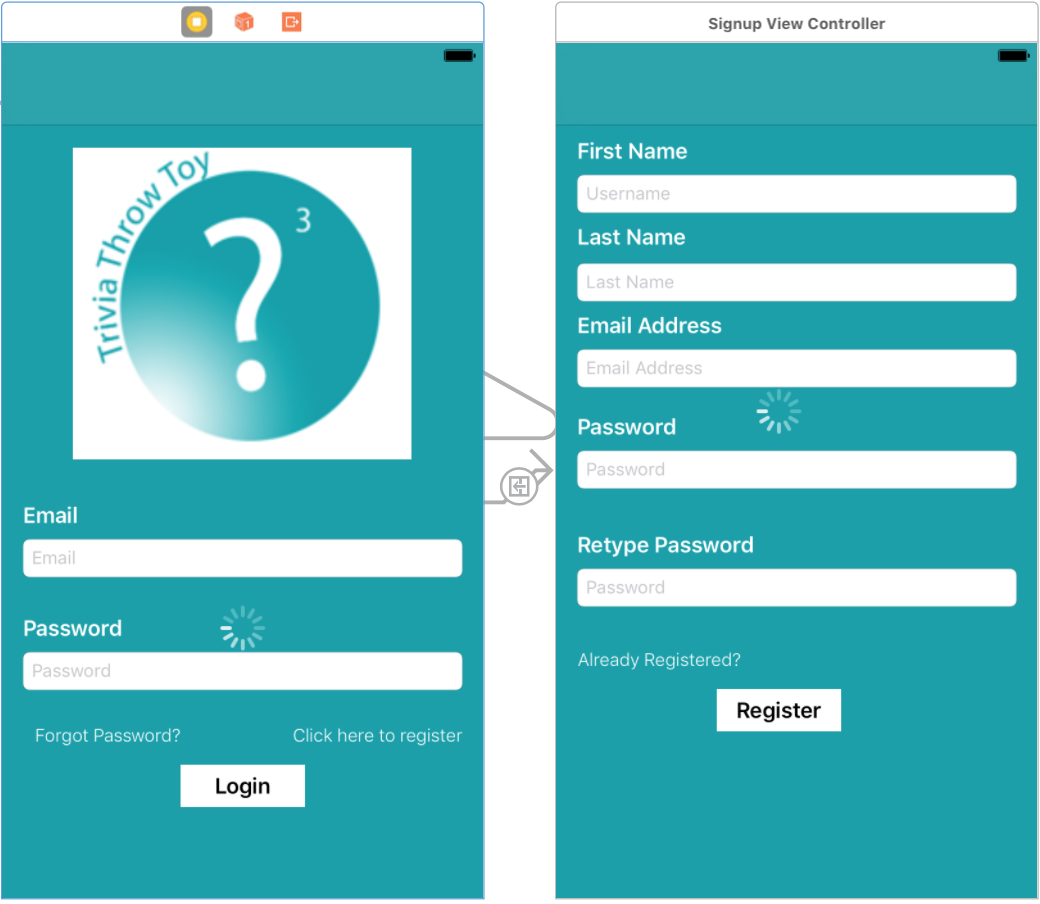
Unless self-explanatory, describe block functions and indicate data or control line contents. Use Visio, Word, or other appropriate electronic drawing tools.



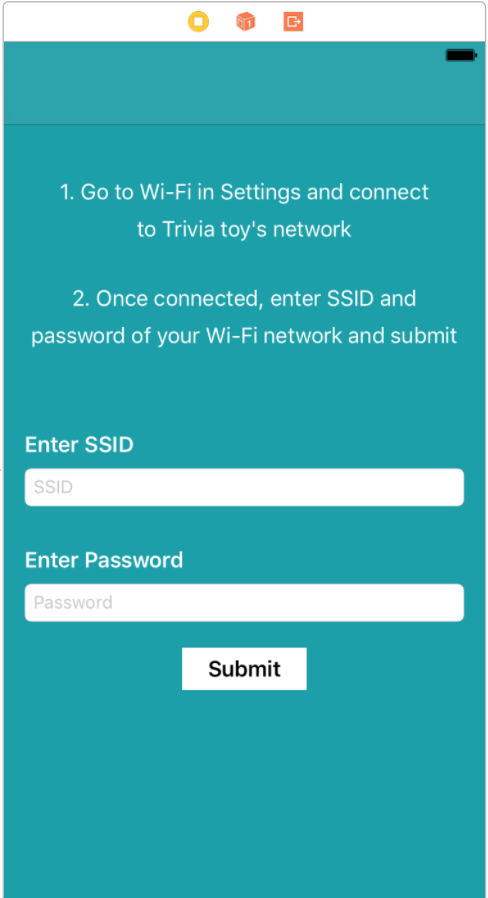
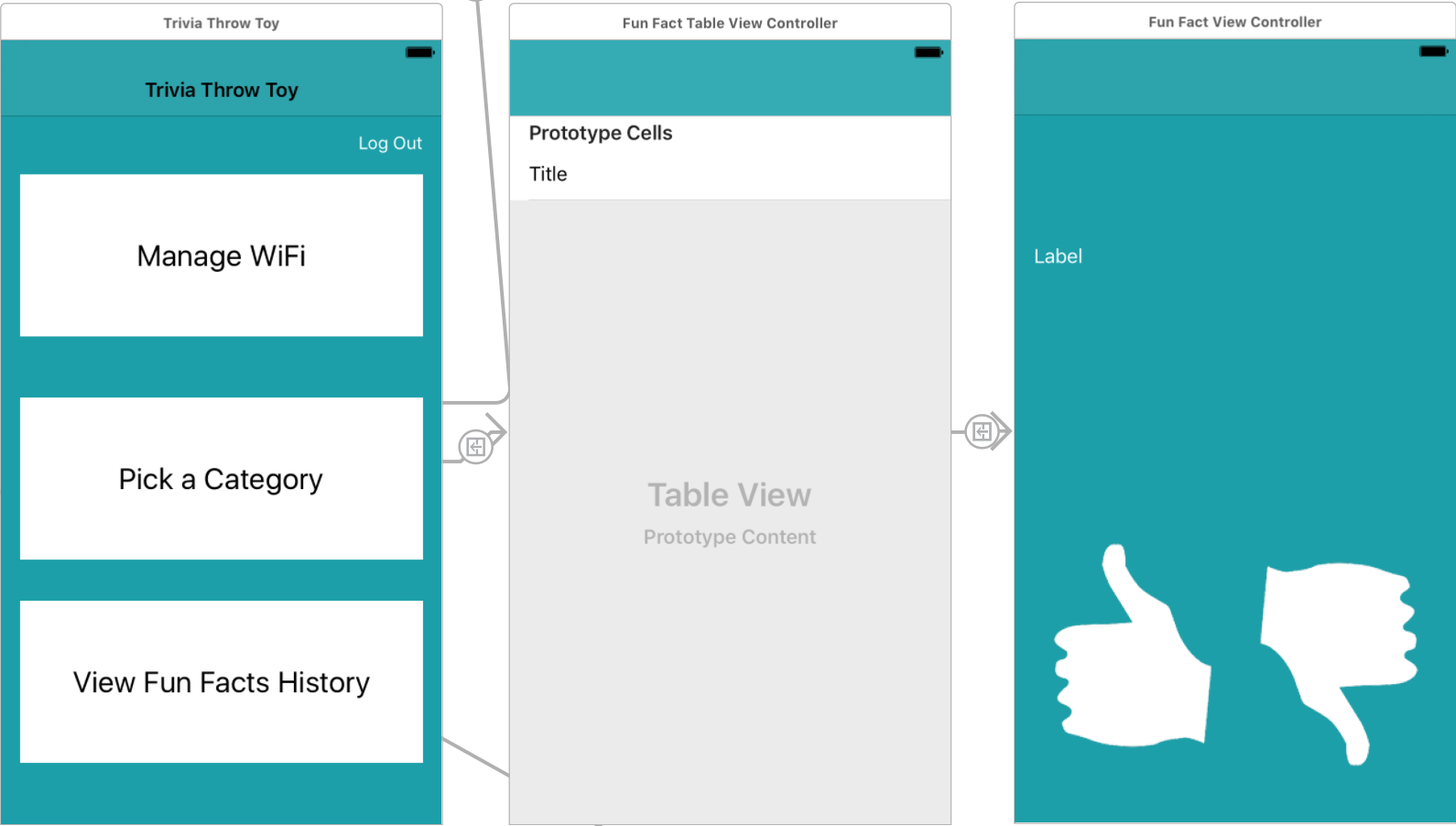


## User interface.

Login and Registration Screen of IOS App

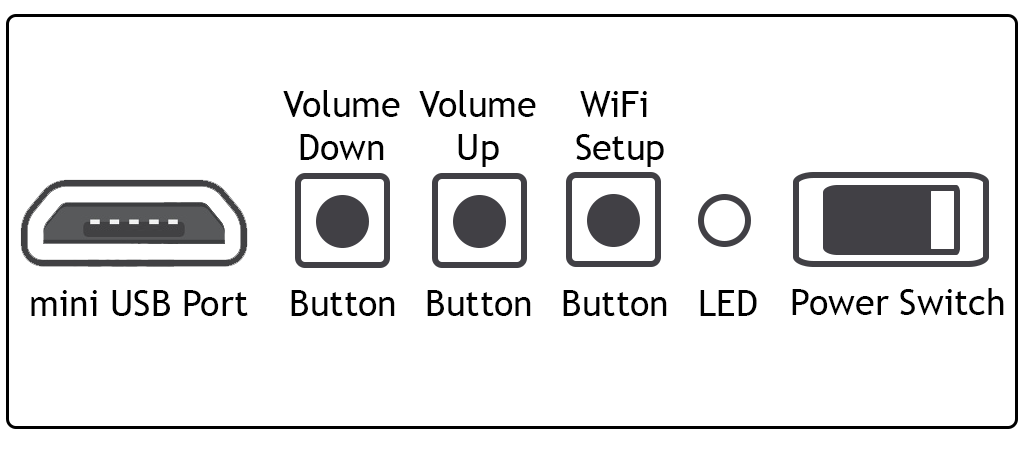


Main Menu And Manage Wifi Screen of IOS App



|  |
| --- |
| Android App: When connection with toy is established, the next step is enabled    Android App: Displays available wireless networks and prompts for a password for selected WiFi    Android App: Displays successfully sent credentials to toy |

Toy User Interface Diagram



## Physical description.

The toy’s shell has holes for the speaker, buttons, LED, power switch, and usb charging port. Inside the shell, is a holder for the PCB. The top and bottom halves have a fitted locking mechanism to hold them together when joined. For the outer lining of the shell, EVA foam is molded to fit the sphere for shock absorbency.

|  |
| --- |
| Top half of shell with speaker holes |

|  |
| --- |
| Bottom half of shell with PCB holder and holes for hardware components for user interaction |

|  |
| --- |
| Assembled model of toy |

## Installation, setup, and support

To set up Trivio, the user first needs to download the mobile application, available for both IOS and Android phones. This can be done by either going to Apple’s App Store or the Google Playstore.

Once downloaded, a new user must register for an account by entering the first name, last name, email address, and password.

Upon successful registration the user can log in to the app.

Users log in to the application with just their email and password that they used during registration.

When the user is logged in, the main menu is displayed. The options available in the main menu are Manage Wifi, View Fact History, and Select a Category.

The next step in setting up the toy is to connect the toy to a wireless network. First, the user must click the WiFi setup mode button on the toy (refer to Toy User Interface Diagram). To confirm the toy is in WiFi setup mode, the LED indicator on the toy will turn white.

In this mode, the toy will emit its own WiFi for the phone to connect to.

The user must go to the phone’s WiFi settings and connect to the toy’s network named “Trivio”. When the phone is connected to the toy’s network, the user can go back to the mobile application and click Manage Wifi. This will redirect the user to a page to enter in the credentials of the wireless network the user wants to connect the toy to. Once the credentials are entered, the user must click the connect button to confirm and send it over to the toy.

The mobile application will then show a page confirming the credentials has been sent to the toy. To determine if the toy is successfully connected, the user can refer to the toy’s LED status indicator. If the LED is green, the toy is successfully connected. If the LED is purple, then the toy did not establish a connection. If unsuccessful, the user can click the WiFi setup mode button on the toy again and repeat the process.

If support is needed, the user can refer to the LED status indicator to determine what state the toy is in. The user can look up the corresponding color meanings in the Toy Status Table.

# 

# Operation of the Project

## Operating Mode 1: Normal Operation

When the power switch on the toy is turned on, the LED on the toy lights up to an orange colour. This indicates that the toy has started and is initializing all the modules.

**Initial Setup (Starting up for the first time)**

1. Press the WiFi Setup button on the toy
2. The LED will light up as the colour White
3. Open the Trivio app
4. Register a new user account by navigating to the signup page and entering your name, email and password
5. After signing in to the application, select Manage Wifi
6. The app will tell you what to do next: Connect the phone to the Toy using the “Trivio” network
7. Once the phone is successfully connected to the toy’s network, a list of WiFi networks will be generated on the app
8. Select the WiFi network you want the toy to connect to, enter the password and select continue
9. Look at the LED for the status of the network connection: Purple - connection unsuccessful, Green - connection successful
10. If the connection was unsuccessful, try moving closer to your router and start the process from step 1 again

**Normal Operation (Already been set up once)**

1. If the toy is in proximity with the WiFi connected to during initial setup, wait for the LED to become Green letting you know the connection to internet was successful.
2. If the toy is in a different location and want to connect to a different WiFi, go through the steps in Initial Setup again
3. If there is no WiFi in proximity, wait for the LED to turn Purple
4. Shake or throw the toy to hear a random fun fact

**Volume Control**

1. Pressing the middle button increases the volume by a level of 1
2. Pressing the leftmost button decreases the volume by a level of 1
3. The volume can only be changed when the toy is not in the process of saying a fact

**Server Request**

1. When the toy is being used, it keeps track of the number of facts played from its local memory
2. Whenever the toy is connected to the internet(WiFi), it immediately downloads the number of facts played to ensure that 50 facts is always stored locally
3. When this happens, the LED colour is changed to Blue

**Battery Status**

1. The toy keeps track of the current battery level
2. When the battery drops to 15% or lower, the LED changes to a Red colour
3. When this happens, connect the charger to the toy. The toy can be used while it is being charged as well

**Fact History and Voting**

1. Open the Trivio app
2. Login if not already logged in
3. Select the Fact History option
4. This will display all the facts that have been heard by the account logged in
5. If a fact is selected, the user can give that specific fact a thumbs up or thumbs down vote

The different LED colours and their meanings are shown in the table below

|  |  |
| --- | --- |
| **Toy Status Table** | |
| **Colour** | **Status** |
| Orange | Toy Start up |
| Red | Battery is <= 15% |
| Green | Connected to the Internet (WiFi) |
| Purple | Not connected to the Internet (WiFi) |
| Blue | Updating from the server |
| White | In WiFi setup mode |
| Pink | Attempting to connect to Internet |

**Implementation Pending**

The app has an option called Categories, which allows the user to select a specific category to get a fun fact from. Even though this option exists on the app, the database does not have the facts sorted into categories yet. When this works, the steps would be:

1. Open the Trivio app and select the Pick a Category option
2. A list of categories available will be displayed and the user can choose one or multiple
3. The next time the toy is shaken or thrown, a fun fact from one of the categories selected will be vocalised

## Operating Mode 2: Abnormal Operations

If the toy is at a location where there is no access to the internet, it is possible for the toy to repeat a fact that has already been heard. The toy plays facts from its local storage, but if it exhausts all these facts, it will reset and play the first fact again.

## Safety Issues

Since the toy is the size of a medium beach ball, it can be harmful if thrown directly and forcefully towards a specific person.

The toy is an electronic device that is quite robust. However, if the toy is thrown with too much force to a wall or is dropped down multiple stories, it is possible for the toy to stop functioning as intended.

The toy can overheat if left in the presence of direct sunlight for long. This can cause the material of the toy to catch on fire as eva foam and plastic is being used for the casing.

Though the outer material is water resistant, the toy should not be operated and charged with wet hands as a short circuit resulting in electrocution could occur, as the case with almost any electronic device.

# Technical Background

**4.1 Physical Toy**

The following components are necessary for the toy’s hardware:

* accelerometer (MPU6050) - to detect a throw or shake
* text-to-speech module (EMIC2) - converts the text files of the stored facts to be played on the speaker
* WiFi module (ESP8266) - enables wireless network connection for the toy to make fact requests to the server and turn the toy into an access point for the phone to connect to and send WiFi credentials
* SD card - stores facts as text files on the toy to be played upon throw or shake
* speaker - plays the fact for the user to hear
* audio amplifier (YL2020) - increases the volume for the fact to be easily heard
* battery - provides power to the toy
* mini usb charging port - allows the user to recharge the battery
* buttons - allows user to control the volume of the toy and put it into wifi setup mode
* LED - acts as a status indicator of the toy for the user to see
* power switch - allows user to turn on and off the toy
* microprocessor (Arduino Mega) - runs the code for the toy

**4.2 Microprocessor Code**

On startup of the toy, the ESP8266 will attempt to make a connection with a WiFi network it has previously been setup to connect to. It will also retrieve the number of facts played since the last server update (meaning how many facts have been played offline and were not replaced because it was not connected to internet), which is stored in a text file on the SD card. It will then attempt to populate the toy’s storage with that number of new facts if connected to wifi by making the HTTP GET request /trivia/{userId}. It will also inform the server which of the downloaded facts have been played by making the POST request /trivia/{userId}/{count}.

For detecting a throw or shake, the accelerometer will poll the x, y, and z acceleration values and if any of them have difference greater than an established threshold from the previous polled value, a throw or shake has been detected. When a throw or shake has been detected, the name of the file for the next fact (the filenames are in incremental order) is retrieved in a text file stored on the SD card. The filename is then passed to a built in EMIC2 function which will read the fact stored from the specified file. The fact will be played on the speaker and the name of the next file will be updated and stored in the SD card. If connected to wifi, a HTTP request is made to the server for a new fact to be downloaded on the SD card. If there is no WiFi, the count for how many facts have been played offline is incremented and stored on the SD card to be read the next time it is connected to WiFi.

The arduino will then continue to poll acceleration values again, waiting to detect the next throw or shake.

**4.3 Mobile Application**

The mobile application makes the following HTTP requests to the server:

* POST: /User/Register - when the user fills in the signup form on the registration page and presses the submit button. The body of the request includes the name, email and password and is recorded via input text fields. A successful HTTP response includes a user ID which is saved on the device in order to save the session for future logins.
* POST: /User/Login - When the user fills in the form on the login page and presses the login button. The user fills in their email and password and it is sent in the body of the post response. A successful HTTP response includes the user ID which is stored on the device as a way to remember the user session for quicker login. Upon pressing the logout button, the user ID is removed from the phone memory.
* GET: /User/History/{id} - when the user opens the View Fact history page to retrieve the facts played on the toy. The response returns all of the facts that have been heard by the user as well as the specific fact id. When a user clicks on one of the facts, they are taken to another page with the fact listed as well as a thumbs up/thumbs down button.
* PUT: /Trivia/Vote/{id}- when the user clicks the thumbs up or thumbs down button of a specific fact, the fact id is sent in the the request URI parameters. Inside of the body of the request includes the user ID, fact ID, and how they voted (1 for up vote, -1 for down vote)

**4.4 WiFi Setup**

To connect the toy to WiFi, the user must send the credentials using the phone application. In order to send the credentials, the toy must be configured as an access point for the phone to connect to. When the phone is connected to the toy’s WiFi, the app can open up a socket server on a specified port in the Manage WiFi page.

The toy’s microprocessor code will detect when a connection made to its own network and obtain the ip address of the device connected. When the ip address is obtained, it will try to make a socket connection using the address and the specified port. The ESP8266 wifi module is also configured so only one device can be connected.

In the mobile app’s Manage WiFi page, when the socket server is created, it will detect if a client joins and establish a 2-way handshake to verify that it is the Trivio toy. Once verified, it will allow the user to continue to the next page which uses Wireless Broadcast Receiver Manager to scan for available wireless networks and displays them in a list. When the user clicks on one of the networks, it will prompt the user for a password.

The toy, after the handshake, will wait to receive the credentials from the socket connection. When the user clicks the button on the app to send over the credentials, the app will also send over the userID associated with the account over the socket connection. The toy then parses what it receives, separating it into userID (which it stores on the SD card for the toy to access and make requests with), the wifi SSID, and the wifi password.

The toy then tries to connect to the WiFi using the received credentials and if successful, will set the LED indicator to green, or if unsuccessful, sets the LED to purple. It then exits out of WiFi setup mode by closing the socket connection and disabling it from being an access point.

The mobile application, once the credentials have been sent will close the socket server and display on the GUI for the user to check the LED status of the toy for successful connection or not.

**4.5 Parsing Facts From Internet**

Every toy is able to connect to a central database of trivia, hosted on Microsoft Azure. This ensures that each toy has access to 1000s of facts, which is not feasible to store locally. Additionally, any changes to the central database are reflected in all the toys in use. As we discover more facts, we can keep updating the database giving users access to more facts.

Currently, we have a parsing algorithm in place which is able to extract trivia from websites. The algorithm requires a website base url and html tag that encloses the trivia as an input, from there on it extracts facts from a page, then automatically moves to the next page on the website which contains trivia. Each trivia is passed through a validation algorithm that tests for length and uniqueness. If the validation test passes, the trivia is not added to the central database.

With minimal changes to the parsing algorithm we are able to extract trivia from multiple different websites. We used this process to currently extract 5000 trivia. As we identify more websites that have trivia, we will continually add more to the database.

# Cost Breakdown

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Costs for Production of Beta Version (Next Unit after Prototype) | | | | |
| Item | Quantity | Description | Unit Cost | Extended Cost |
| 1. EMIC2 | 1 | Text To Speech module | 63.35 | 63.35 |
| 2. MPU6050 | 1 | Accelerometer | 4.97 | 4.97 |
| 3. SD Card | 1 | Micro SD Cards | 5.99 | 5.99 |
| 4. EB-B600BUB AMZ | 1 | Rechargeable battery | 10 | 10 |
| 5. ESP8266 | 1 | WiFi module | 3.75 | 3.75 |
| 6. Adafruit Micro Lipo | 1 | Battery charger module | 10.59 | 10.59 |
| 7. SD Card Reader | 1 | SD Card Reader Module | 12.26 | 12.26 |
| 8. Resistors | 6 | Surface mount PCB | 0.5 | 3 |
| 9. EVA foam | 2 | Outer shield material | 1.42 | 2.84 |
| 10. Arduino Mega | 1 | microprocessor | 37 | 37 |
| 11. Voltage Converter | 1 | Converts voltage to 9V | 4.94 | 4.94 |
| 12. PCB | 1 | PCB | 28.90 | 28.90 |
| 13. Buttons | 3 | Buttons on toy | 0.25 | 0.75 |
| 14. LED | 1 | LED on toy | 0.12 | 0.12 |
| 15. Headers | 150 | Headers for modules | 0.01 | 2.06 |
| 16. Speaker | 1 | To hear the fact | 2 | 2 |
| 17. Audio amplifier | 1 | To amplify the fact being spoken | 3.36 | 3.36 |
| Beta Version-Total Cost | | | | 195.88 |

The EMIC2, ESP8266, MPU6050, Micro Lipo Charger, SD Card Reader and Voltage converter are all modules needed for the toy. These modules are wired together on a PCB that is fabricated. The eva foam is used as the outermost layer for the toy to prevent it from damage if accidentally dropped.

When ordering the PCB, it comes it quantities of 3. In the table above, the quantity of 1 PCB has been calculated and included.

Currently, the toy shell is being 3D printed at EPIC for free. However, if the toy was to go into production, an alternative would be needed for mass producing the shell which would affect the budget.

# Appendices

## Appendix A - Specifications

|  |  |
| --- | --- |
| **Specification** | **Value** |
| Ideal operating temperature | 32° to 95° F |
| Operating Volume | 30-80 dB |
| Maximum fact retrieval time | 1s |
| Offline fact cache | 50 |
| No of facts at launch | 4,899 |
| Maximum fact length | 140 |
| Battery Life | upto 4 hours |
| Charging time | 2 hours |
| Battery Capacity | 2600 mA |
| Connectivity | 802.11a/b/g/n/ac Wi‑Fi with MIMO |
| Ports | micro usb |
| Outer material | Eva foam |

## Appendix B – Team Information

**Jacob Dansey**

[jdansey@bu.edu](mailto:jdansey@bu.edu) (703) 994-0146

Jacob Dansey is a Computer Engineer major in Boston University graduating in May 2017.

**Changshuo Fu**

[fcs1994@bu.edu](mailto:fcs1994@bu.edu) (484) 479-4670

Changshuo Fu is an Electrical Engineer major in Boston University graduating in May 2017. He has experience in programing languages C, Python and MATLAB. He is a former engineering intern at ASML-Brion (Shenzhen,China). He loves music and sports. In free time, he likes to play the flute and practice taekwondo.

**Christine Low**

[clow@bu.edu](mailto:clow@bu.edu) (347) 831-1080

Christine Low is a Computer Engineering major in Boston University graduating in May 2017. She has a concentrated software engineering background with experience in web development, android mobile applications, 3D modeling software, and programming languages such as C++, C, C#, Java, Python, and MATLAB. She is a former software engineering intern at Microsoft and will work there full-time upon graduation.

**Urvashi Mohnani**

[umohnani@bu.edu](mailto:umohnani@bu.edu) (857) 265-8205

Urvashi Mohnani is a double major in Computer and Electrical Engineering at Boston University graduating in May 2017. She has experience in programming languages C++, C, C#, Java, Python, and MATLAB. She is a former intern of Plum Voice and a former coordinator of the Artemis program, teaching incoming high school females about computer science and engineering. She also currently works for BU Engineering IT.

**Neil Sanghrajka**

[neils95@bu.edu](mailto:neils95@bu.edu) (867) 263-1231

Neil Sanghrajka is Computer Engineering major in Boston University, graduating in May 2017. He has strong experience in server side development using Node.Js and the MEAN stack for web development. He is a former intern at ITC Infotech, San Jose and a teaching assistant at BU. His technical interests are Geospatial databases and Mobile