LSU College of Engineering Department of Mechanical & Industrial Engineering

College of Engineering School of Electrical Engineering & Computer Science

To Predict ▶ To Design ▶ To Perform

ME, ECE Capstone Design Programs

Team #50: Eye-Gaze Interface for Facebook

Jacob Bechler, Nicholas Charles, Austin Copper, Rory Hector

St. Lillian ACADEMY

Background

- Julian is a young man with cerebral palsy who uses eye-gaze technology to communicate.
- Eye-gaze devices track your eyes and map their movements to selections on a screen.
- Social media platforms, such as Facebook, offer a means of communication for those who are unable to talk on the phone or visit someone personally.

Objectives

- Our objective is to provide Julian an interface for Facebook that he can use on his current eye-gaze device, the Accent 1400.
- This interface should recreate the core functionality of Facebook.
- In addition to capturing the essence of Facebook, this interface should be highly optimized for use with eye-gaze technology.

Engineering Specification

Measurable Specifications	Units	Target values	Explanation	Results
Time until fatigue	Minutes	60	The small eye muscles fatigue quickly using eye gaze devices	45
Time until frustration	Minutes	60	Frequent errors lead to frustration and dissatisfaction with the product	>45
Time from selection to result	Milliseconds	500	Eye-gaze interfaces need to support real-time browsing	40 - 8000
Rate of mistakes	Errors per minute	< 5	Precision with eyes is hard to maintain, organize design to aid in this	Varied, typically <5
Time spent fixing mistakes	Seconds per mistake	< 10	It mitigates frustration if mistakes can be easily corrected	Too varied to measure
Size of interface selections	Percentage of total screen	> 5	Button sizing will be tested to optimize interface for eye-gaze usage	Varied, all >5
Selections per screen	Natural numbers	< 10	Want to avoid clutter and too small buttons while not requiring many pages	<7
Task completion time	Percentage of time relative to old app	75	The optimizations should improve completion times on all tasks	42.1
Percentage of tasks that require assistance	Percentage	0	The user should be able to browse without ever needing help	0

Methods

Embedded Browser

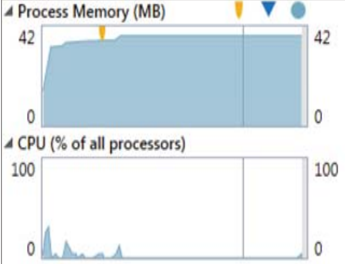
- We will use Visual Studio's *WebBrowser* to display data otherwise inaccessible due to restrictions on Graph API.
- The browser will be controlled externally with large buttons and other interface components.

Graph API

- Facebook's Graph API is the primary way to externally retrieve and post data to Facebook.
- We will use it in our C# code for all data access that graph API allows.
- Otherwise, we will use the WebBrowser to access the data.

Testing and Validation

CPU & Memory



Big-O Analysis

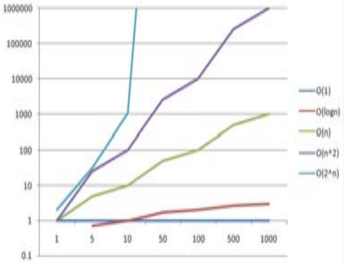
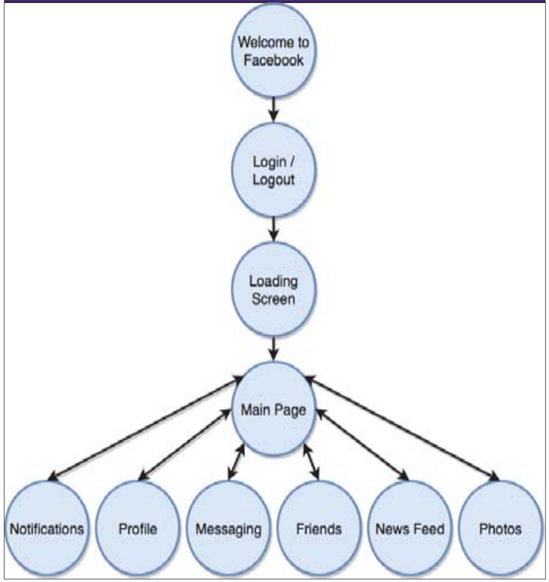


Image above obtained from: https://www.daveperrett.com/articles/2010/12/07/comp-sci-101-big-o-notation/

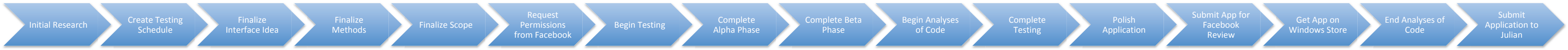
Program Flowchart



Code Samples

```
private void Button_Click_3(object sender, RoutedEventArgs e)
{
        var fb = fb_client();    ← Establish connection to Facebook
        dynamic result = fb.Get("/me/posts");
        var temp = fb.Get(result.data[0].id + "/comments?fields=from");
        for(int i = 0; i < temp.data.Count; i++){          ↑
                Comm1.Text = temp.ToString();
        }                                   Collect Facebook data using Graph API
}
```

```
var fb = fb_client();
dynamic result = fb.Get("/me/posts");
var id = result.data[0].id;
dynamic reply = new ExpandoObject();
reply.message = reply_box1.Text;
fb.Post(id + "/comments", reply);  ← Use Graph API to post directly to Facebook
```



Initial Research ▶ Create Testing Schedule ▶ Finalize Interface Idea ▶ Finalize Methods ▶ Finalize Scope ▶ Request Permissions from Facebook ▶ Begin Testing ▶ Complete Alpha Phase ▶ Complete Beta Phase ▶ Begin Analyses of Code ▶ Complete Testing ▶ Polish Application ▶ Submit App for Facebook Review ▶ Get App on Windows Store ▶ End Analyses of Code ▶ Submit Application to Julian

Sponsor: Elissa McKenzie

Advisor: Dr. R Vaidyanathan