Final Report Cloud Group

Michael Meding, [Michael\_Meding@student.uml.edu](mailto:Michael_Meding@student.uml.edu)  
Ramon Meza, [Ramon\_Meza@student.uml.edu](mailto:Ramon_Meza@student.uml.edu)  
Tylor Reeves, [Tylor\_Reeves@student.uml.edu](mailto:Tylor_Reeves@student.uml.edu)

Website Link: <http://umlsofteng.web44.net/>

# Table of Contents

[Table of Contents](#h.pg4qbj4sc72d)

[Abstract](#h.g137doz94k5s)

[Description](#h.n4k3oiucipj9)

[General Overview](#h.it0r4vak7dux)

[System Overview](#h.mvf1b2khms9f)

[Design Considerations](#h.q46avp9r6sk7)

[Assumptions and Dependencies](#h.ao8cx57pif6w)

[General Constraints](#h.xdbo2sulvdyv)

[Goals and Guidelines](#h.opo3rauh29w1)

[Development Methods](#h.6ph68scourde)

[Architectural Strategies](#h.qnlok7c0tmr4)

[System Architecture](#h.u4p7x98wcush)

[Policies and Tactics](#h.l2uy8i80mlnd)

[Detailed System Design](#h.xmn7ypdcqe3m)

[System Installation and Execution](#h.ssc299dwjeqc)

[Glossary](#h.l3056d83o479)

[Bibliography](#h.im4om6vqo4z)

# Abstract

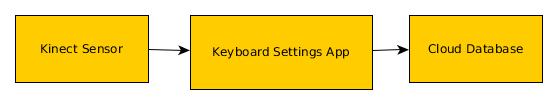
For our SWEN project we developed a Kinect based keyboard emulator. The original goal of this project was to create a system that allows users to be more physically active at their desks. This system works by mapping the user’s movement with a predefined set of keystrokes. Once the keystrokes have been captured, they are broadcasted to whatever application the user wishes to use. The usage and movement data is also sent to the cloud where words per minute, accuracy, and other usage statistics are calculated. There is also a companion app installed on phones which allows the user to view their data in the cloud.

## Description

### General Overview

* The purpose of this project is to create a system that will allow users to be more physically active at their desks
* The target audience for our product is anyone with a Kinect. This means that we will be targeting both children and adults.
* Provide references for any other pertinent documents such as:
  + Related and/or companion documents
  + Prerequisite documents
  + Documents which provide background and/or context for this document
  + Documents that result from this document (e.g. a test plan or a development plan)

### System Overview



# Design Considerations

## Assumptions and Dependencies

Our system has several dependencies in order to function as intended. They are:

* An Xbox or Xbox One
* A correctly calibrated Kinect sensor
* Up to date Xbox operating system
* A registered Xbox Live account
* A reliable Internet connection

## General Constraints

This project has several design and hardware constraints based on the dependency of many factors. These factors include, but are not limited to:

* Hardware constraints are high because this software will only run on one platform with one specific set of hardware
* The keyboard will also require the user to have a lot of space to interact with the controller
* To be able to log into the system, the user will be required to have an internet connection. Not having one will limit the software
* This software will be compliant with all Kinect software regulations
* The database will be simple, only storing users’ data with their keybindings and layout
* Only basic security will be used to secure user passwords for their keyboard accounts
* This software will need to be fast enough to run in real time to allow the user to type at a reasonable rate. This will require a fast processing of input, which will put a strain on both the Xbox and its memory
* The software must be tested and validated to be able to run for extended periods of time, as well as handle any possible combination of valid or invalid user input
* The software must be able to take note of and report all errors in log files for debugging purposes and to ensure the software is working as intended
* The software must be thoroughly tested for all types of bugs that might affect an end user’s experience

## Goals and Guidelines

The purpose of this project is to implement a working keyboard, which only requires a Microsoft Kinect. This will be supported by an intuitive UI with simple to understand motions that link to characters mapped out on keyboard. Development, to the extent called out by the above requirements section, is all that is planned. The end product will run at several public display kiosks or in local Microsoft stores. This will allow the most exposure for both the Kinect technology and our end product.

We do not plan on integrating any kind of game associated with the keyboard. We also do not plan on implementing detailed tutorials with this release because the design for the interface should be made simple enough for anyone to understand quickly.

## Development Methods

Our development method for this project was the tried and true agile method. We performed two week long sprints to get the project to where it is now. These sprints consisted of daily meetings where we talked about what we had done as well as what was to be done next by each person. This really helped us stay on track with the project and get things done quickly and efficiently.

# Architectural Strategies

* **The following programming languages and technologies were used**:
  + PHP: The easiest and most common way for our project team to interact with the MySQL database.
  + HTML: The standard markup language to create web pages with Bootstrap.
  + Javascript: The standard scripting language to create web pages with Bootstrap.
  + CSS: The standard style sheet language for use with Bootstrap.
  + Bootstrap: An easy to use framework that made building an attractive website easier.
  + phpMyAdmin: The most popular and easy to use software tool to manage a MySQL database using PHP.
  + MySQL: The most common database management system that the project team members were familiar with.
* **Reuse of software/technologies:**
  + Bootstrap: Using the bootstrap framework, we were able to create a slick and appealing website in much less time than if we had created the website from scratch.
* **Plans to extend software life:**
  + Depending on the popularity of the application and the importance of the requests from users, we may implement features that were not originally planned to enhance our software after release. Any bugs found in the application after release will be fixed, but as of now, there are no specific plans to extend or enhance our software beyond the original release date.
* **Relevant diagrams:**
  + Please see the detailed system design section as well as the link to assignment 3 in the bibliography for user interface paradigms, software paradigms, and other relevant diagrams.
* **Error detection and recovery:**
  + All necessary cloud system methods will check for errors and, if found, pass them to an error handler which will create an appropriate log file in a protected directory so that administrators may view it for debugging purposes.
  + If the cloud subsystem has crashed or become stuck in a deadlock, the system will attempt to either recover or, if not possible, restart the system.
* **Memory/storage management policies:**
  + The structure of the MySQL database provides restrictions on the type and size of column values, so as to conserve space.
  + The communication between the cloud database and kinect application will be optimized and minimized, so as to conserve application memory usage and reduce bandwidth usage.
* **Distributed data or control over a network:**
  + The data will be distributed over the cloud, data aggregation, and sensor subsystems so that all components are integrated and communicate with each other.
* **Concurrency and synchronization:**
  + Github was used for this project as the version control system to keep our project up to date.
* **Communication mechanisms:**
  + Communication for this project took place in the forms of a google group page for discussion between all subgroups, as well as daily meetings between cloud subgroup members both in person and over the instant messaging application Google Hangouts.
* **Management of other resources:**
  + PHP was used by users to both retrieve and manipulate the contents of the MySQL database.

# System Architecture

The cloud subsystem consists of the following smaller subsystems:

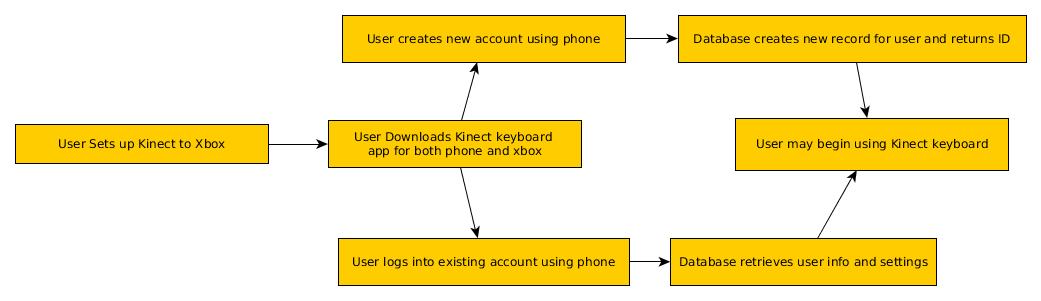
* **User registration system:**
  + This system is responsible for handling the registration and login of users and the changing of user settings via the cloud website.
* **Phone application system:**
  + This system is responsible for connecting to the cloud database and validating user login credentials when a user attempts to login to the companion phone application.
* **Xbox application system:**
  + This system is responsible for connecting to the cloud database and validating user login credentials when a user attempts to login to the Xbox One Kinect application.
* **Kinect keyboard usage statistics system:**
  + This system is responsible for handling data sent to the cloud database by the Xbox One application and storing it in the database. This system must also handle requests from the cloud website and phone application to send this data so that it may be displayed to users.
* **Cloud website usage statistics systems:**
  + This system is responsible for showing users their Xbox One Kinect application usage statistics on the cloud website.

# Policies and Tactics

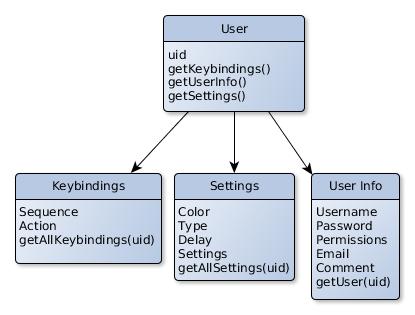
* **Choice of which specific product to use:**
  + PHP and MySQL were used as they are the most common ways to manage a database and were the easiest for our team to use.
* **Coding guidelines and conventions:**
  + All of the code that the team wrote was properly documented to help viewers understand the code better.
  + Industry standard naming conventions were used to eliminate any confusion about variable and function names.
* **Plans for ensuring requirements traceability:**
  + Github was used as a version control system to document all code changes.
  + Documentation was given at three different points during software development which can be used to trace the original and current requirements.
* **Plans for testing the software:**
  + Aside from manual testing and validation by our team, the cloud system will have an error handler that documents all errors that the system might encounter during use.
* **Plans for maintaining the software:**
  + Any bugs mentioned by users or noticed by our team will be fixed as promptly as possible.
  + Suggestions by users will be taken into consideration, depending on their popularity.
* **Hierarchical organization of the source code into its physical components (files and directories):**
  + ├── app  
    │ └── js  
    │ └── users.js  
    │ └── validateChange.js  
    │ └── validateForm.js  
    │ └── php  
    │ └── mysql\_connect.php  
    │ └── mysql\_login.php  
    │ └── mysql\_logout.php  
    │ └── mysql\_register.php  
    │ └── mysql\_settings.php  
    ├── assets  
    │ └── css  
    │ └── animate.css  
    │ └── styles.css  
    │ └── img  
    │ └── blur-city.jpg  
    │ └── js  
    │ └── bootstrap.js  
    │ └── jquery.js  
    ├── about.php  
    ├── dashboard.php  
    ├── index.php  
    ├── login.php  
    ├── original.php  
    ├── register.php  
    ├── settings.php  
    └── README.md
* **How to build and/or generate the system's deliverables:**
  + The cloud system consists of a website and a database, both of which are constantly accessible and only need to be uploaded to a web server to be used.

Each particular policy or set of tactics employed should probably be discussed in its own subsection, or (if it is large or complex enough) in a separate design document (with an appropriate reference here of course).

# Detailed System Design



* **Classification**



* **Definition:**

The cloud provides users with a website that allows them to interact with the cloud database which stores all user information and settings.

* **Responsibilities:**

The cloud component must, through the cloud website, allow the user to register an account with the cloud database and store/change user settings for the kinect sensor application so that the user will not have to enter their settings each time they launch the kinect application.

* **Constraints:**

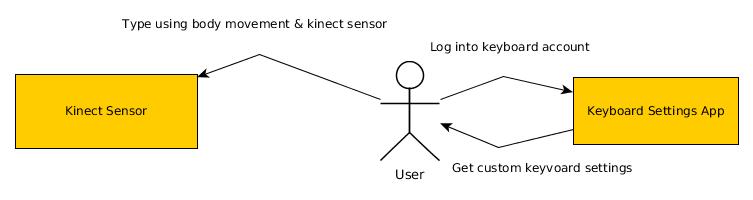
As the application will be running in real time, both our free hosting site and the application user must have a sufficient internet speed so that data can be transmitted between the end user and cloud database quickly enough that there will be little to no latency. As we are using free hosting, our storage is limited and we must ensure that data storage is optimized so that no unnecessary space is wasted.

* **Composition:**

**Website:** The cloud website will be the main interface that users will interact with. The website must be able to properly communicate with the other subcomponents of the cloud system.

**Database:** The cloud database will be a MySQL database administered using phpMyAdmin, and will store all user information such as username, email, password, keybindings, and user settings. The database will communicate with the cloud website so that users may update their application settings.

* **Uses/Interactions**



* **Resources:**

**Bandwidth**: While not a physical constraint, the speed at which the kinect sensor application is able to communicate with the cloud database will be limited by the bandwidth of both the end user and the database host.

**RAM:** Noting that all Xbox One consoles come with a predetermined, non-upgradable amount of RAM, the amount of memory that the kinect application uses while communicating with the cloud database must be minimized so that the Xbox One OS and kinect application itself will always have enough memory to function without latency. Memory deadlocks should be handled by ensuring that no process will have exclusive access to a partition of memory and by process termination: terminating processes in order of ascending importance until the deadlock is resolved, and then having the application attempt to recover, if possible.

**Xbox One Storage:** The Xbox One comes with a fixed sized HDD, which is 500GB (1TB on limited models), that cannot be upgraded through normal means. Noting that most end users will have many other applications taking up space on their Xbox One HDD, we must do our best to minimize the size of the final kinect sensor application, as well as minimize the amount of user-specific data that is stored on the end user’s Xbox One.

**Hosting Storage:** Our free host provides us with 1500MB of storage for our database and server files. This amount should be more than enough for the current implementation of our project.

* **Processing:**

**Website:**  The website will present end users with an HTML page with the option to either login with an existing account, or create a new account. The website will validate user credentials or add new user credentials to the cloud database using PHP statements and SQL queries. After the user has successfully logged in, the user will be presented with an interactive HTML page including options to change their password or email, add keybindings, update their settings, and view application usage data, among other miscellaneous options. These changes will be first validated to ensure that the content entered by the user does not contain any illegal characters, and then updated in the cloud database using PHP statements and SQL queries. If the current user of the website happens to have administrator permissions, the user will be presented with potential extra options such as resetting/changing another user’s password, enabling debugging options, and otherwise modifying the cloud database.

**Xbox One Application:** Upon starting the Xbox One Kinect application, users will be prompted to either login to an existing account, or create a new one using the cloud website. Login information supplied by users will be validated by the cloud database using SQL queries. Users will be encouraged, but not required, to enter their Kinect keyboard settings such as keyboard type, color, and keybindings upon account creation. If no additional settings are entered by the user, the user will be given default settings. Once users have logged into their cloud account on the Xbox One application, the application will initialize the kinect keyboard using the user’s settings which were pulled from the cloud database. When a user closes the Xbox One application, detailed statistics on their usage will be sent to the cloud database so that users may view the information either using the cloud website or phone application.

**Phone Application:** Upon starting the phone application, users will be prompted to either login to an existing account, or create a new one using the cloud website. Once the login information has been validated by the cloud database, users will be taken to a page detailing statistics regarding their kinect keyboard usage, such as words per minute, accuracy, and total usage time. This information will be pulled directly from the cloud database.

**Exception Handling:** Upon the occurrence of an exception, an error log file will be generated with all pertinent details regarding the exception and sent to a protected directory on the cloud website server via FTP.

* **Interface/Exports**

The interfaces provided to the end user by the cloud website include a page to register/login, a page to view keyboard usage statistics, and a page to change settings. The inputs on the register/login page include username, email, other account settings, such as keyboard type and color, which will be stored in text format in the cloud database, and a password which will be stored in encrypted format. These inputs will be filtered and handled by PHP statements that interact with the SQL database. Inputs on the settings page include keyboard type, keyboard color, and keybindings, all of which will be stored in text format and handled by PHP statements. Keybindings may be entered by the user whereas keyboard type and color may be chosen from a list of possibilities. The data usage interface will not contain any inputs, and serves only to show data collected about the user’s keyboard application usage.

# System Installation and Execution

Installation of our software is quite simple. As the project will be hosted on the Xbox marketplace the only thing that is required of the user is to go to the marketplace and download our Xbox app.

From there the user must only input their username and password or create a new account and the keyboard will be loaded with the default configuration. Some short explanations will be shown and then it is up to the user to use the keyboard with other apps.

# Glossary

**Kinect**: The camera device attached to the Xbox that tracks the motions associated with our product.

**Xbox**: The gaming device which will be running our code.

**Keybindings:** Saved combinations of keys that, when recognized by the kinect sensors, will cause the system to perform a certain operation.

**HDD:** A physical hard drive disk used for the storage of data.

**RAM:** Random access memory, used by an application to execute processes.

**OS:** Operating system, software that manages a devices hardware and software.

**Github:** The most popular version control system, used to keep files up to date with other team members and to document/manage all changes.

# Bibliography

Assignment 1 - Feasibility Report: https://github.com/UMLproject/Software\_Engineer\_91.411\_3/tree/master/3\_Cloud/Document/Assignment1-Feasibility-Report.docx

Assignment 2 - Scrum Prototype Report: https://github.com/UMLproject/Software\_Engineer\_91.411\_3/tree/master/3\_Cloud/Document/Assignment2-ScrumPrototype.docx

Assignment 3 Diagrams: https://github.com/UMLproject/Software\_Engineer\_91.411\_3/tree/master/3\_Cloud/Document/Assignment%203%20Diagrams

Adapted from IEEE documentation standards (particularly IEEE-1016 and IEEE-830)

Copyright © 1994-2015 by IEEE

*Permission is hereby granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies*.