###### Matthew Thomson

###### Victor Estopinan

Kevin Delamo

A Unique System to Teach the necessary skill of touch typing

Typing Teacher: Requirement Elicitation

2016

# **1. INTRODUCTION**

Typing Teacher provides a flexible system for teachers to provide students with a fun and engaging way to improve his or her typing skills.

In section 1.1, the current system will be discussed. Section 2 will be dedicated to describing the requirements and constraints of the proposed system. Section 3 describes the system models, specifically the use case model in section 3.1.

## 1.1 Current System

The current system in place is a static, rigid course that does not allow the teacher to create their own courses. Teachers are restricted to a predefined system. In addition, the current system maintains a system destined to make students see learning as a chore, more than as a fun experience. Words like course, assignments and other drab terminology keep students feeling like learning is a nothing more than a process they are put through rather than an engaging activity.

# **2. PROPOSED SYSTEM**

## 2.1 Functional Requirements

### 2.1.1 Game Play

* The system will display a typing game to the user that assesses the user's typing accuracy and speed.
* The user will be able to interact with the game by typing
* The system will display results in common metrics such as Words Per Minute (WPM) and accuracy percentage.
* The system will support two types of game modes:
  + Phrase Mode: The system will display a phrase to complete.
  + Time Trial Mode: The system will display a count down timer and unlimited phrases to complete. -The system will allow the use of items for various purposes such as stopping the timer during time trial mode.

### 2.1.2 User Functionality

* The system will allow users to be able to see historical and aggregated statistical information about specific game, aggregated campaign, and overall results.
* The system allows registered users to create of custom campaigns and levels.
  + Campaigns are a collection of levels.
    - Campaigns must have at least one level.
  + Levels a set of game parameters and passing requirements.
    - Levels must be part of a campaign.
* The system allows teachers to create specialized custom campaigns.
  + These specialized campaigns can have
    - Deadlines
* The system allows registered users to export and import campaigns.
* The system will allow users to see their inventory of items.

### 2.1.3 User Roles

* The system will distinguish between anonymous users, registered users, student users, and teacher users.
* The system will allow anonymous users to:
  + Play the game and receive immediate results.
* The system will allow registered users to:
  + Do anything an anonymous user can
  + View historical and aggregated statistical data about game results
  + Create custom campaigns
  + Retain an inventory
* The system will consider registered users as students while in a teacher's class campaign.
* The system will allow teacher users to:
  + Do anything a registered user can
  + Create teacher class campaigns
  + Assign student users to class campaign
  + Can track student progress and game results in the class campaign.

## 2.2 Nonfunctional Requirements

### 2.2.1 Usability

* Any user who has basic computer literacy will be able to use the system without instructions or the user manual.
* The system will provide visual cues to guide the users without any previous knowledge.
* Any user with English proficiency will be able to understand the system

### 2.2.2 Security

* The system will support a user name and password for each registered user.

### 2.2.3 Performance

* The system will process typed keys and input by the user in less than a second.
* The system will display correct results after each play.
* The system will save results correctly for each registered user.
* The system will save data for all registered users after each play.
* The system will save data after each level is created when creating a campaign.

### 2.2.4 Supportability

* The system will accept upgrades/patches.

## 2.3 Constraints

* Must use a database
* Must be web based
* Must have at least three user roles
* Must be completed before 12/1/2016

# 3 SYSTEM MODELS

## 3.1 Use Case Models

## 

Figure 3.1-1 Caption: Diagram of Typing Teacher's Use Case Models

### 3.1.1 Account Use Cases

#### Register Account

* Actors: Anonymous
* Entry Condition: Not signed in
* Flow of Events:
  1. User clicks register
  2. System prompts user for Account name and password (required)
  3. User enters required information ^i
  4. System validates and generates user account.
  5. Includes use case signInAccount
* Exceptions:
  1. Account name must be unique.
* Exit Condition: Registration Submitted

#### Sign In Account

* Actors: Registered users
* Entry Condition: Not Signed In
* Flow of Events:
  1. User clicks Sign In
  2. System prompts user to enter credentials, account name and password.
  3. User enters credentials.
  4. System signs in user ^i
* Exceptions:
  1. If credentials cannot be authenticated, system prompts user with an error of incorrect username or password.
* Exit Condition: Signed In

#### Sign Out Account

* Actors: Registered user
* Entry Condition: Signed In
* Flow of Events:
  1. User clicks Sign Out
  2. System signs out user
* Exit Condition: Not Signed In

### 3.1.2 Campaign Use Cases

#### Create/Edit Campaign

* Actors: Registered User
* Entry Condition: Authenticated User
* Flow of Events:
  1. User clicks Create Campaign
  2. System shows options
  3. User sets available options for the campaign:
     + Starting items
     + See use case "createLevel"
  4. System adds created game to the campaign and asks user if the user needs to create more levels.
     + If so, system prompts user if this is a branch point.
       - If so, create a two level branch.
       - Repeat 2.
  5. User finishes campaign. ^i
  6. System saves campaign and returns user to dashboard. ^ii
* Exceptions:
  1. If the campaign does not have any levels, the system will display an error telling the user that the campaign requires at least one level to finish.
  2. If the user cancels without saving, the system will not save the campaign and returns the user to dashboard.
* Exit Condition: Finish making campaign or cancels.

#### Create/Edit Class Campaign

* Actors: Teacher User
* Entry Condition: Authenticated Teacher
* Flow of Events:
  1. Extends use case createEditCampaign.
  2. User may add a deadline to the campaign. ^i
  3. User may add a user white list to the campaign.
     1. The system will verify the user to be added exists. ^ii
     2. The system adds the user to the white list.
     3. Upon creation of the campaign, the system will notify all users on the white list that they have been added to this campaign by the teacher.
     4. The system adds this campaign to all whitelisted users' campaign lists.
* Exceptions:
  1. If the date selected is in the past, the system prompts the teacher with an error stating that the deadline has already passed.
  2. If the user does not exist, the system prompts the teacher with an error stating that the user does not exist.
* Exit Condition: Finish making campaign or cancels.

#### Create/Edit Level

* Actors: Registered users
* Entry Condition: Must be Signed In, Must be creating or editing a campaign
* Flow of Events:
  1. System displays the level options and level completion requirements
  2. User Updates: ^i
     + options:
       - Game Mode
       - Timer length or Passage length
     + completion requirements:
       - required accuracy percentage
       - required words per minute
* System saves level and returns.
* Exceptions:
  1. User inputs invalid input. System prompts user to correct input with helpful tips.
* Exit Condition: Level created or saved

#### Export Campaign

* Actors: Registered users
* Entry Condition: Authenticated Registered User, User must have created campaigns.
* Flow of Events:
  1. User clicks to export campaign.
  2. System displays the campaign code.
* Exit Condition: None

#### Import Campaign

* Actors: Registered users
* Entry Condition: Authenticated Registered User
* Flow of Events:
  1. User clicks to import campaign.
  2. System prompts user for campaign code.
  3. User inputs campaign code. ^i ^ii
  4. System displays campaign.
  5. User can accept or deny the campaign.
     + Accept: System adds the campaign to the user's campaign list
     + Deny: The system returns -Exceptions:
* Exceptions:
  1. If the campaign code is invalid, the system will display an error telling the user that the campaign code is invalid, and allow additional attempts.
  2. If the campaign code is valid, but the user is not on the whitelist, then the system will display an error to the user stating that they do not have access to the campaign.
* Exit Condition: None

### 3.1.3 Gameplay Use Cases

#### Play Game

* Actors: All users
* Entry Condition: User starts game
* Flow of Events:
  1. User starts game
  2. System provide game type
  3. (Generalization) User interacts with game. (playPhraseMode, playTimeTrialMode)
  4. User replays or quits game
* Exit Condition: User finishes game

#### Play Phrase Mode

* Actors: All users
* Entry Condition: Game started
* Flow of Events:
  1. System displays passage based on user
  2. User types the phrase as accurately as possible ^i
  3. System analyzes and displays results to user. If the user is registered, the system saves the results.
* Exceptions:
  1. Typos are indicated in red. Possible to correct typos.
* Exit Condition: Phrase Complete

#### Play Time Trial Mode

* Actors: All users
* Entry Condition: Game started
* Flow of Events:
  1. System displays passage and timer based on user and starts timer.
  2. User types the phrase as accurately and quickly as possible ^i
  3. If there is still time remaining display another phrase. Repeat 1.
  4. When time expires the system analyzes and displays results to user. If User is registered, the system saves results.
* Exceptions:
  1. Typos are indicated in red. Possible to correct typos.
* Exit Condition: Timer Complete

### 3.1.4 Information Display Use Cases

#### Access Campaign

* Actors: Registered User
* Entry Condition: Authenticated user
* Flow of Events:
  1. User clicks campaign
  2. System displays
     + Campaign Name
     + Completion Percentage
     + Campaign Levels
       - Aggregated data about the level
     + Detailed run history ^i
       - Aggregated data about the run
* Exceptions:
  1. If the user does not have any past runs, then the system does not display any.
* Exit Condition: None

#### Access Class Campaign

* Actors: Teacher User
* Entry Condition: Authenticated Teacher
* Flow of Events:
  1. Extends use case
  2. System also displays
     + Registered Users that are in the class
     + Overview of class progression
     + Details of individuals progression
* Exit Condition: None

#### Access Dashboard

* Actors:
* Entry Condition:
* Flow of Events:
  1. User clicks dashboard
  2. System displays the following information for the user
     + Campaign List ^i
       - Campaign name, progress, history.
     + Regular game history
     + Create Campaign
     + Import Campaign
* Exceptions:
  1. If the user does not have any class campaigns, then the system does not display any.
* Exit Condition: None

#### Access Teacher Dashboard

* Actors: Teacher User
* Entry Condition: Authenticated Teacher
* Flow of Events:
  1. Extends use case accessDashboard.
  2. The system also displays:
     + Class Campaign List ^i
       - Abbreviated aggregated data: progress, completion percentage, etc.
* Exceptions:
  1. If the user does not have any class campaigns, then the system does not display any.
* Exit Condition: None

# 4 GLOSSARY

* **Campaign**: A set of related levels in which the levels must be traversed from start to end in order to complete the set of levels.
* **WPM (Words per minute):** The average number of words typed per minute.