Project Brief

Document Information

Project name:	Typing Trainer
Date:	26/02/21
Author:	TeamSoftwareProject Group 8
Owner	Jason Quinlan
Document code:	
Version:	2.0

Key Stakeholders

Major Stakeholder	Notes
Product Owner	Jason Quinlan
Developers	Raffaele, Shane, Niamh, Cormac, Pavel

Notes

Our Typing Trainer will utilise Markov Chains to generate the sample text for users to practice on. This will allow them to practice their typing skills with word sets that are more likely to be experienced in practical use. Similar products use quotes and excerpts from text as their practice sets, which might not be representative of real life use. Our goal is to provide a complete user experience, with the ability to compare your stats against yourself, your friends, and the wider user base. Other competitors such as Type Racer or Keybr offer some of the functionality, but not all. Type Racer allows one to compete against their friends, but doesn't offer test sets that represent real life typing, while Keybr has the ability to generate representative practice sets, but lacks some of the competitive aspects we wish to include.

Our application offers users multiple categories from which the Markov Chains will generate text sets making the game more captivating and appealing to all kinds of users. Online games will be measured by how quickly and accurately a player can complete the text set. For our initial launch we intend to integrate user profiles and a 'friends' feature which lets you compare your game statistics with other friends. This also holds great potential for expansion through additional social features in future releases.

Background:	Initially we were interested in creating a highly interactive mobile application that processed user data and allowed users to compare their stats with their friends. We identified typing as an activity that is part of our day to day life, especially with remote learning, and decided to base our application around this. We reviewed our initial model and decided that a web based platform is more appropriate than mobile given our short time frame but kept with the idea of making it highly interactive and competitive. A mobile-based application can be considered for a future release since smartphones and tablets are becoming increasingly popular than traditional laptops/desktops. There also appears to be a similar gap in the market for mobile-based typing training applications. After discussing potential features our team agreed we all enjoyed games that returned feedback and statistics based on your performance - a great example of this is lichess which is an online multiplayer chess game which reports potential flaws in your technique after you complete a game. Based on this observation we agreed to implement a visual display of game statistics such as average words typed per minute, word accuracy etc - this will be implemented in private user profiles. A potential future revision of this would include historical data which would track your progress.
Main Goal:	To create a web-based Typing Trainer that fills the gaps in current offerings such as stat comparison and diverse yet practical word sets which are split up into multiple categories. The site will give the users an option to log in or to simply practice in 'guest mode'. Users that are logged in will have their game history saved, the ability to add/remove friends and compare their historical game statistics with their friends , and the option to practice in single or multiplayer mode. They can view their game stats such as average words per minute and accuracy. Options will include basic targeted letter sets for people learning to touch type. A secure account management system with minimal personal user information storage with important information (such as passwords) hashed using sha256.
Desired Outcomes:	The aim is to launch a fully interactive typing application that is able to support multiple concurrent users. Ideally we hope to launch the application on the public domain, however due to the short timeframe for the project we will focus more on completing the application rather than launching it. The layout of the webapp should be responsive, support modern browsers and all non-mobile screen sizes. Receive useful feedback and bug reports from external testers/users through a beta release. This will aid in driving the project forward in terms of amending/improving functionality. Implement all features outlined in the ‘Must have’ and ‘Should have’ columns of the MOSCOW prioritisation.

Constraints and Assumptions:	• It will be run on Modern browsers. (We lack the resources to test our code on legacy hardware and software). • 7 week deadline - strict schedule with low slack. • Developers have other responsibilities - college and work. • Developers are comfortable with the main development tools: mySQL, Python 3, HTML and CSS. • Time must be dedicated to learning new frameworks which include: Flask, bootstrap, pyChart.js. • Time must be dedicated to testing each application component, performance/responsiveness of games and hosting security/availability. • Package dependencies will be necessary, compatibility could be an issue across platforms. • Functionality does not require mobile interaction. • System differences between team members can limit testability/choice of technology.
Interfaces:	None
Project Approach:	• The project will be developed in house using the Python web-framework Flask, with a SQL database. • The mySQL database will be launched using AWS RDB service. • Testing will be done locally in hopes of launching it on an online web server. • Figma will be used to create a prototype of the front end interfaces giving us a concrete design to aim towards. • Developers have agreed to use the Google Python Style Guide as the common styling approach. • Will set out weekly goals during meetings and keep everyone updated on progress through slack. • Revisions will be made on a weekly basis. • We considered reformatting our code by taking an Object Oriented approach for ease of future development, however on further revision we decided against this as it would require too much change too far into the project. • We are using Githubs inbuilt project management tools to keep track of individual tasks and using Gantt charts to visualise our progress. • The Netsoc cloud service will be used for hosting a beta release of the application.
Project Product Description:	Typing Trainer using computer generated text to create naturalistic test sets.

Outline Business Case

The Typing Trainer aims to close the gap in the market offered by already established typing aids. It has been identified that other vendors lack a unified platform - some have realistic typing sets while others have an in-depth analysis/feedback system but there appears to be a scarcity of applications that offer a combination of all of the outlined functionality.

The benefits of this project would include an opportunity to centralise this functionality and allow users to improve their typing skills in a fun, responsive and interactive manner. A realistic and practical neuralistic text set will be generated meaning users have an opportunity to practice on text that is applicable in everyday life, from emails to college assignments.

Risks and dis-benefits include the highly competitive market. Typing aids are very common and take many forms from online educational standalone games, to full on dedicated platforms such as Keybr and Type Racer. Privacy is a paramount concern and great emphasis must be placed on the security of account information and user statistics. The costs would include hosting the application in the cloud and scaling performance to allow multiple concurrent users as the application gains popularity, however at launch free tier platforms will be used. Other risks include the tight 7 week time deadline with the possibility of delays in development.

Project Objectives

	Target	Tolerance
Scope	Markov chain to generate text, Responsive interface to show typing progress, Stored user accounts and typing statistics, Statistic processing, Visual prompts (highlighted words, graphs, progress charts etc), Web based platform	Simple text sets, Simple interface - only displays text with no visual cues, Guest Users only, no friend networking Statistics only for most recent run Offline prototype - site is not launched but fully functional otherwise
Time	7 weeks	No tolerance
Cost	Cloud Hosting	Free tier cloud hosting
Quality		
Risks	Requirements change Hardware failure Size underestimate Application security	Time constraints may not allow new requirements Minimise functionality

	Staff illness	
Benefits	Implement all features discussed to a high standard Familiarise ourselves with new technologies	Develop a working product

Project Management Team

Role	Reports to	Appointee
Data Interpretation - **pymySQL for db interaction** - **pyChart.JS for data visualisation** - **Flask**	Product Owner	Shane
Functionality - **Markov Generation** - **Web Scraping**	Product Owner	Raffaele
Database/Authentication - **mySQL** - **Flask frameworks** - **Password Hashing**	Product Owner	Pavel
UI/UX - **Figma prototype** - **Front-end (CSS, Javascript)** - **Dark Mode**	Product Owner	Niamh
Functionality - **Private Pages** - **User Profile Pages**	Product Owner	Cormac

All developers are responsible for documentation.

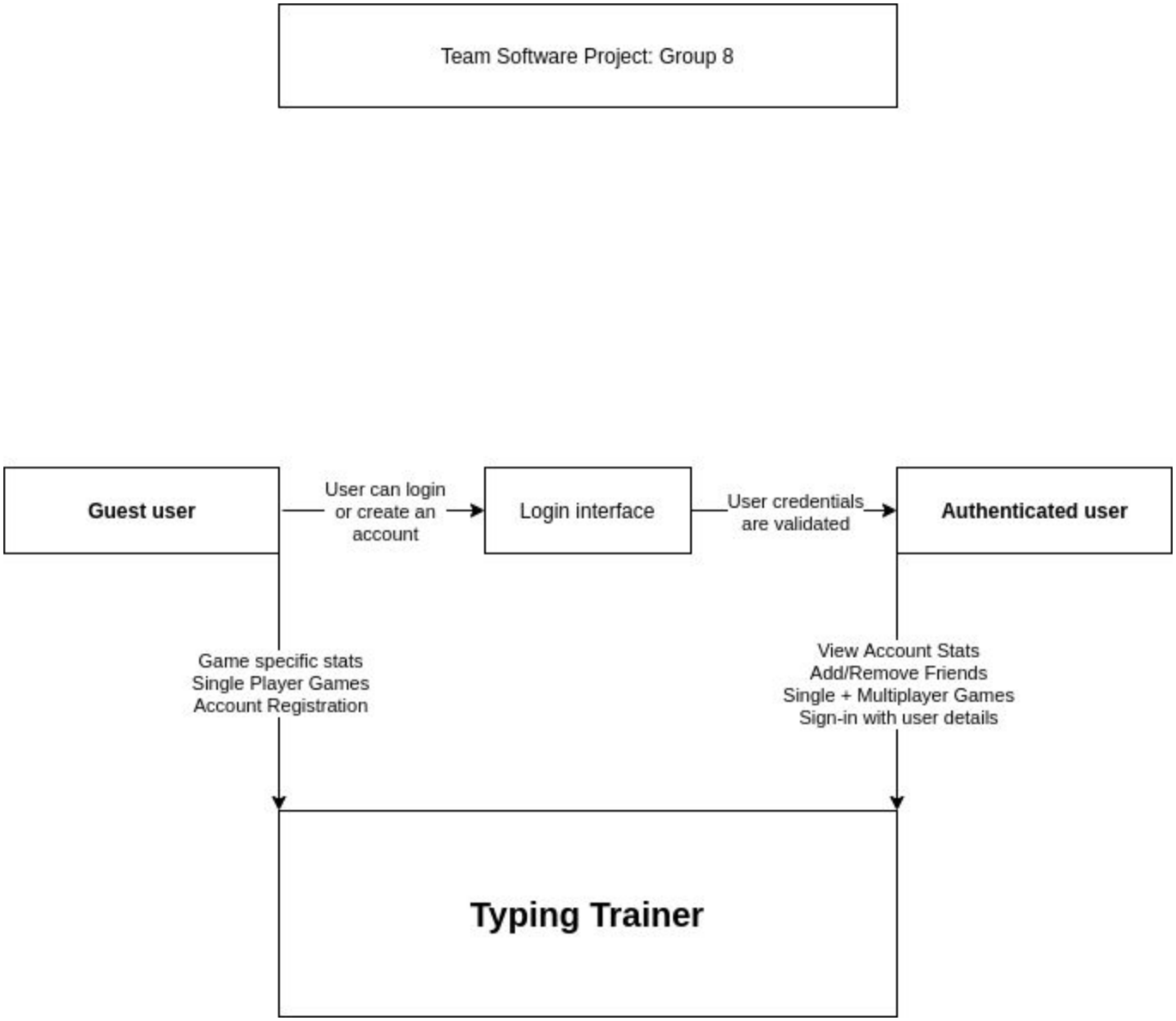
MOSCOW prioritisation

Must have	Should have	Could have	Will not have
Stored user accounts & statistics	Dark Mode	Alternate Keyboard layouts	Freemium Model - Purchasable Categories, Limited Energy/Plays
Visualisation of user statistics	Comparison with other users statistics	Category usage breakdown	Account recovery system
Realistic text sets with different categories		Account profile pictures	
Online matches		Confirmation of registered emails	
Low latency for keyboard input		Admin/Super-User Accounts	
		Social Media integration	
		Historical data	

Architecture Diagrams:

Context Diagram:

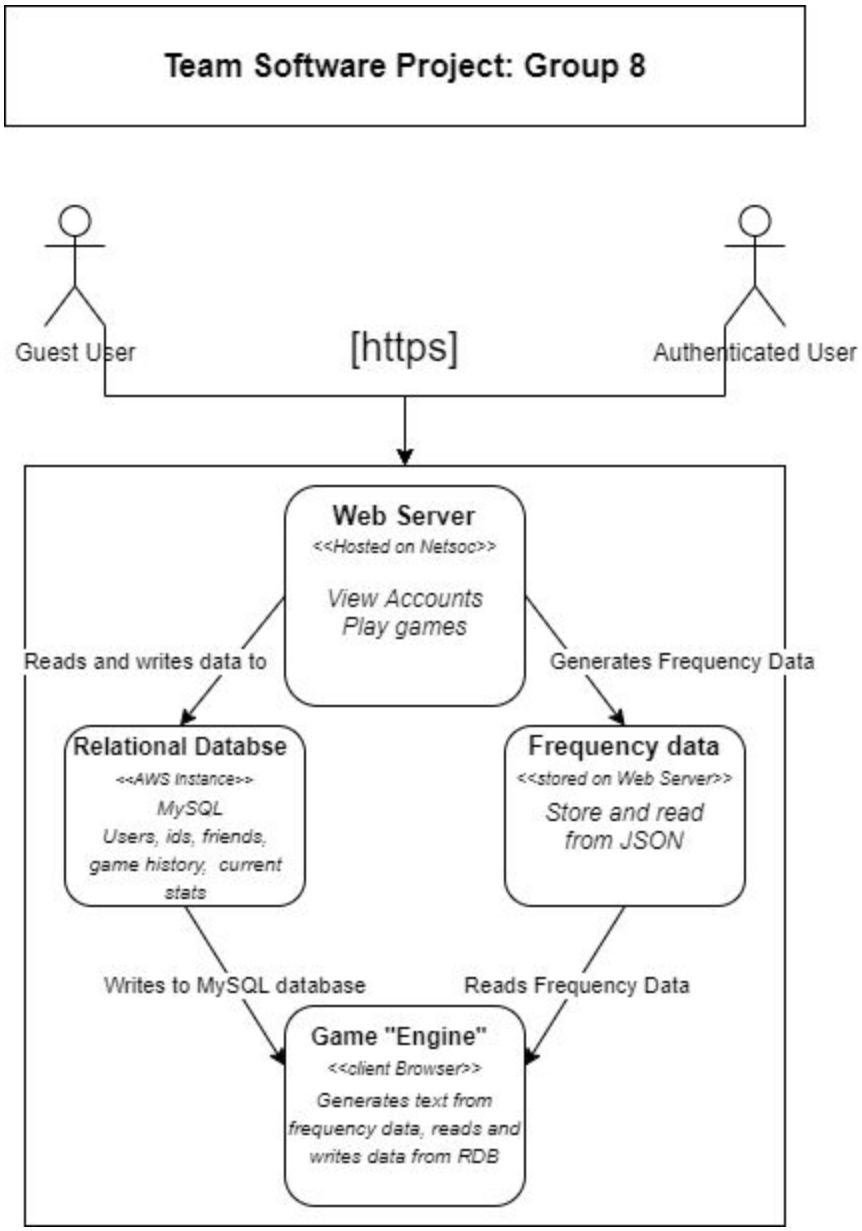
This context diagram is a simple illustration showing who the main users interacting with the system will be and how they will be able to interact with it. In the case of our website, the main users will be split into two categories, guest users and authenticated users, as a guest user you will be able to play single-player games but only get the stats for each individual game, there will be no facility to get aggregated stats. Authenticated users on the other hand will have full access to single and multiplayer games, and will be able to see their stats of the lifetime of their account, for individual games, and compare the two between them and their friends.



Container Diagram:

This container diagram simply lays out the various technologies that will be used to implement our website.

- Our web server will be hosted through UCC Netsoc, on a Ubuntu virtual machine, with frequency data also being stored on the server as well.
- All our relational databases are hosted on an AWS instance.
- The game engine will be run on the client's browser, with interfaces built-in to the Web Server and AWS instance.



Gantt Chart:

The Gantt Chart below will be used to track our progress weekly. It helps us to see what everyone is working on and how the project is coming together. It can point out areas where we are lacking, or if too many people are allocated a task that can be done by just one or two people.