

Online Typing Training Application

COMP8045 – Major Project

--- A00000000
1-26-2019

Table of Contents

1. Student Background.....	3
1.1. Education	3
1.2. Work Experience	3
2. Project Description.....	3
3. Problem Statement and Background.....	4
4. Complexity	7
5. Scope and Depth.....	8
6. Test Plan.....	8
6.1 Unit testing.....	8
6.2 Regression testing.....	9
6.3 Integration testing.....	9
6.3.1 User registration.....	9
6.3.2 Typing practice.....	10
6.4 System testing.....	10
6.5 Acceptance testing.....	10
7. Methodology.....	11
7.1 Methodology & Approach.....	11
7.2 Technologies Involved.....	11
7.2.1 Project management.....	11
7.2.2 Development.....	11
7.2.3 Deployment.....	13
8.1 Client.....	13
8.2 Server.....	14
8.3 Database.....	14
9. Innovation	14
10. Technical Challenges.....	16
11. Development Schedule and Milestones	20
12. Deliverables.....	22
13. Conclusion and Expertise Development	22

14. References	22
15. Change Log.....	23

1. Student Background

My name is - -. I decided to change my career in 2012. I quit my previous chemistry lab job to attend BCIT CST program in September 2013 as a full-time student, then got my diploma at the end of 2015. Then I found a software support and develop job in Schneider Electric. I support and develop web application and databases. Since September 2016, I started my BTECH program as a part time student. I am in both wireless and mobile and human computer interaction major. So far, I only have one elective course and this project left to get the bachelor's degree. I

1.1 Education

BCIT BTECH part time, Sep 2016 to present

BCIT CST full time, Sep 2013 to Dec 2015

Kwantlen College Sep 1999 – April 2001

North Delta Senior Secondary May 1997 – June 1998

1.2 Work Experience

Schneider Electric, Burnaby, BC Canada, software developer and support, Dec 2015 to present

AeroInfo, IT support, Richmond BC Canada, May 2014 to Dec 2014 (BCIT Co-op)

Acme Lab, Vancouver BC Canada, Nov 2010 – Aug 2013

Bestbuy customer service call center, Surrey Canada, 2008 – Oct 2010

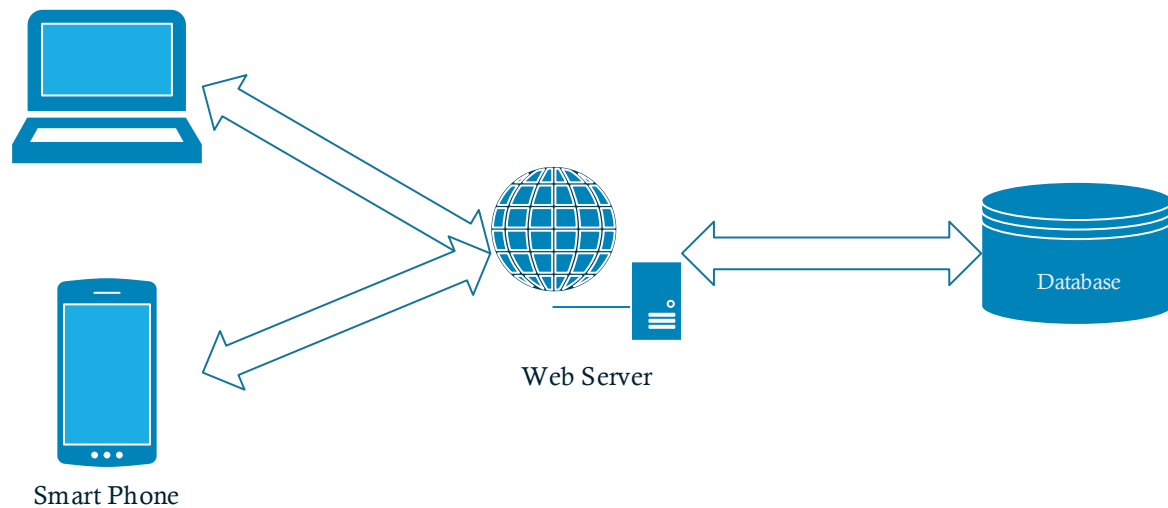
Wesco Industries, Shipper Receiver, Burnaby, BC Canada, 2007 - 2008

Startec, telemarketing, Vancouver BC Canada, 2005 – 2007

2 Project Description

This project is for BCIT BTECH's project course COMP8045. This is my individual project from proposal, design, implementation, test and deployment.

The goal of this project is to create a free online typing software to let users, especially elementary school students to practice their typing skills. This project will mainly consist of a web site client which can be access from both PC and mobile devices, a Microsoft .Net Core web server and an SQLite database.



3 Problem Statement and Background

The problem my application will solve is that users can customize the text they use to practice typing, and more importantly, this application can act like a smart typing coach which can provide the most efficient practice according to user's recent typing performance.

When I teach my daughter typing, I think it is a good idea to choose some sample essays online to let her type, so she can read good essays during learning typing. There are many software for learning typing, free or paid. I spent a lot of time to search an app to allow me to upload an essay to let my daughter type, but I did not find any. My daughter is in elementary school. she is learning writing. At home, English is not our primary language, so I always want my daughter to do more reading. I believe if she reads more good essays written by native English-speaking students, she can gradually learn the way how native English speakers express their meaning in written and improve her writing skills.

I found many school kids' sample essays online, I told my daughter to type them in Microsoft word while she's reading original essay text from another web page. However, it is difficult to check what she types is right or wrong. My daughter also told me that it is hard for her to switch attention between two windows during typing.

The following screenshot shows what I need for my daughter's typing software (green shows correct, red shows wrong), but users should be able to upload text to practice, not just use the static practice words design by the website.



The first problem I want to solve is that I think all users should be able to upload text under their accounts and type what they want to practice. Users, especially school kids can combine typing and reading to make learning more efficient, so that they can gradually improve all of their typing and writing skills.

One reason that kids do not want to learn is that they do not see the learning is useful. Youth education is different from adult education. Kids expect to see positive feedback much faster than adult, otherwise kids will lose patient. If I tell my daughter that good typing skill is very helpful for your high school and college study, and even for your future work, then she may say that, why do I have to learn it now? I design this typing application to give my daughter and other school kids as potential users a much faster reward not only for their typing but also for their writing skill. It will encourage them to learn more. This is my education rational behind the project.

The typing application I am going to develop does not have any grammar checking functions. All essays that teachers select to store in this application have no grammar mistakes. As a developer, I assume the typing trainers or teachers should read all essays before uploading them. Student users will only get good sample essays.

For example, this is a sentence in an essay. "I went to a park with my parents last week." The grammar is correct, and this is what students will see when they open this sample essay. When they type this sentence, they also read it, so they will read correct grammar. It is much better than typing a lot of unrelated words, for instance "happy train goose we sky run" Students can still use these unrelated words for learning typing, but they waste the opportunity to read good English at the same time.

Let go back to the good sentence example "I went to a park with my parents last week." When a student types this sentence, the typing application will not check grammar. It only checks the typing. If the student types "I went to a psrk with my psrents last week.", then the application will detect two "a"s were typed wrong, and it will show "I went to a park with my parents last week." The mistyped letters will turn red. The following shows how it works.

Before typing, what the student will see words are in grey:

I went to a park with my parents last week.

When they type, they will see the following after each letter entered. (I omit the space character in this example)

I went to a park with my parents last week.

I w ent to a park with my parents last week.

I we nt to a park with my parents last week.

I wen t to a park with my parents last week.

I went to a park with my parents last week.

I went to a park with my parents last week.

... ..

I went to a park with my parents last week. (Student enters "a" wrong)

I went to a park with my parents last week.

I went to a park with my parents last week.

... ..
I went to a park with my parents last week. (Student enters “a” wrong)

I went to a park with my parents last week.

I went to a park with my parents last week.

... ..
Finally, the student will see

I went to a park with my parents last week.

As what it shows above, this is not an essay typing software as Microsoft Word. This is a typing training application. It does not check and fix grammar mistakes. It checks the key pressed against the practice text and highlights mistakes. The practice texts can be prepared and uploaded by users, so all users do not have to use the same practice texts provided by the typing training application.

The second problem I want to solve is the user typing statistics. Many typing software provide general typing result score such as word per minutes, and accuracy rate. What I want my daughter to focus is where she often makes mistakes. As a user, my daughter needs a more detailed report to show recently, for example a week, which letters she types wrong and the mistake percentage of each letter. For example, in recent 10 practices, 15% of her mistakes is about letter “w”, 10% is about letter “o” etc. Then I can target train her letter “w” and “o”. I will select words with more “w” and “o” to let her practice. My learning rational is that learners should spend more time on what they are not good at rather than repeating what they have already known very well. Keep typing the pre-selected text repeatedly in hope to correct some habitual mistake is very inefficient. When I collect enough words with “w” and “o”, I will upload all these words into my daughter’s account, which is the first problem I am going to solve, so she can focus on this customized practice.

The third problem I want to solve is based on the solution of the previous two problems. If users have many saved texts in their accounts, my typing training software can suggest them which saved text is the most efficient practice to improve their typing skill according to the type errors they made recently (Users can set the time range). I will develop an algorithm to analyze, calculate and suggest the best practice text based on users’ detailed typing reports. For example, I have saved many essays to practice typing in my account, then the typing software can suggest me to practice an essay because this essay have more letters that I typed wrong recently.

Overall speaking, I believe that each person’s learning experience is different. A general typing application with the same pre-selected text or designed lessons for all learns are not efficient. Each learner should have his or her own training practices according to the problem they encounter during learning. These three problems I am going to solve in my application are based on long time observation of my daughter’s learning process. It distinguishes my typing software from others. After I finish this project, I will design a specific typing training for her by using this application. Other parents or school teachers can also use my typing application to design unique training practice for kids. I also plan to contact my daughter’s school if the school needs a free typing software and if they have specific requirements, so I can keep developing a school version.

My typing application has its own limitation. It is for users who have already know the basic typing such as which fingers control which keys. My typing application does not include the most basic typing lessons because there are many online typing websites provide these features. My application will focus on customized typing training to improve typing skills more efficiently, and use typing to help students to improve essay writing skill.

4 Complexity

The technical complexity of this project consists of the follows:

Web server API: The Web API should handle the user authentication according to different roles.

Code first database control: it should handle multiple database schemas, for example there is a schema for user and role information, and another schema for users' typing record related data. The Object Relational Mapping (ORM) used in this project is Microsoft Entity Framework Core, which is not the same as what I used previous pure Windows version Entity Framework.

User experience: this is another difficult part. The user interface should be attractive and intuitive, otherwise school kids won't like it. The front web site must be responsive to both PC and mobile devices. Users can use mobile devices to login and see their typing reports. I need to learn Angular before start implementing this project.

Typing records analysis algorithm: This algorithm should efficiently calculate and produce a report for each individual user about the specific errors they typed and search existing stored text to suggest the best practice which target users' weaknesses.

The management complexity of this project consists of the follows:

Risk management: There are some technologies I need to learn before using, for example Karma, the front-end unit test framework. I plan to use Angular Animation for the web interface. I also need to learn it first. Another risk is that this is an individual project, so I do not have teammates to discuss if I encounter difficulties and there will be no code peer review to find potential bugs.

Schedule management: This is the first time I develop such a big application individually. Some technologies I must learn. Some technologies I used before in my full-time job are based on senior co-workers' setup and configuration, so I might encounter many troubleshooting problems to delay the progress. My plan to solve this issue is to create some small test projects first to try these technologies, then start my major project.

Diploma students cannot solve this problem because they do not have the knowledge to individually manage and develop a project with such a scope and requirements. In BTECH's Technical Issues in Software Development and Management Issues in Software Development courses, I learned Agile methodology and what to do to prepare the risks before and during the development process.

Diploma students mostly focus on implementing the functionalities, but they do not have enough knowledge to think in detail about the software maintainability and development progress

management. I graduated from CST program, so I know what I could do at that time. BTECH students can see the project from a high-level management point of view, for example use Agile Scrum methodology to keep delivering workable software. In one of my specialties, User Computer Interaction, we learned not just user interface (UI) design, but also user experience (UX) design principles. Sometimes a better user experience can make a software more competitive than just providing more functionalities. I also learned about SQLite, client-server access and responsive design in wireless and mobile course. These knowledges will be used in this typing software.

Diploma student do not pay too much attention to unit test in their projects. Most projects, at least in the CST program I took, students mostly use manual test. BTECH students can do much better in automated test to make the software more robust.

5 Scope and Depth

The minimum scope of the project is to create an online typing software which allows users to upload text into their accounts to type and then generate reports to show the error percentage according to each key on the keyboard.

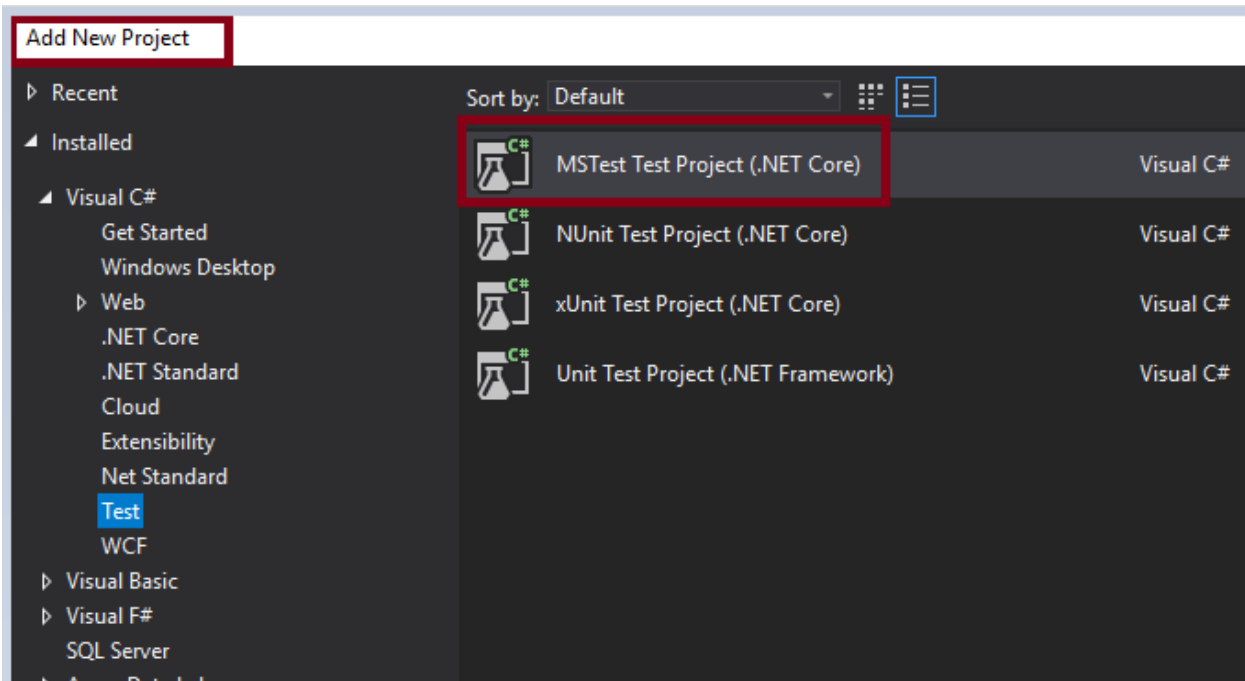
The depth of this project includes the follows:

- The web site is responsive to both PC and mobile devices, so users can review their typing reports from difference devices. Users can also upload text from mobile devices.
- A user friendly and intuitive interface.
- Sophisticated algorithm to analyze the weak point of a user's typing skill according to the typing records and then suggest exercises which target their weak points from pre-saved text.
- Include different roles of users, for example teacher and student, because this application is potentially can be used in elementary school education.
- Teachers should be able to assign typing tasks to students and then receive students' typing report, so teachers can also manually assign students target exercised according to their typing records.

6 Test Plan

6.1 Unit testing

The back-end web server will use "MSTest" to do unit testing. An "MSTest" project will be created for each project under the entire solution. (Note that "project" here means a project under Microsoft Visual Studio solution. The solution which contain all source code of both front-end and back-end will have multiple projects).



In MSTest project, Moq will be used for mocking dependencies.

All Web API controllers, services and repositories will have at least one unit testing method associate with.

The front-end unit testing will use Karma. I need to learn Karma first. When the front-end component in Angular framework is created, each component has its own HTML, CSS, TypeScript file for logic and another TypeScript file for unit testing.

Unit testing will be run after each method implementation.

6.2 Regression testing

Regression testing will be handled by unit testing and manual testing. Each time before committing code to version control, both unit testing and manual testing must pass.

6.3 Integration testing

Integration testing is based on all unit tests pass and no other error found in manual testing. Integration testing will run after one entire use case is finished. For instance, a user types one minute and the type record is sent to web server and then saved into database. This user case includes front-end operation to the back-end logic. After each layer's component is done (controller, service, repository) for the entire use case process, an integration test will run to test different cases.

6.3.1 User registration

A user provides an email and a valid password

1. A user pending activation record is created in User table in the database
2. A confirmation email is sent to user's email to get confirmation.

A user clicks the link which sent from this online typing application

1. This user's pending activation record is active in the database
2. User gets a message on the web page to show this account is active.

6.3.2 Typing practice

A registered user signs in the online typing web site

1. This user can practice typing in this application
2. This user can save customized text under this user's account

An unregistered user opens the online typing web site

1. This user can also practice typing in this application
2. This user cannot save customized text so the text will be lost after closing the web page

A registered user finishes a typing practice

1. A simple report which shows this practice result on the web page
2. The typing record will be saved into database for this user

An anonymous user finishes a typing practice

1. A simple report which shows this practice result on the web page

User successfully uploads text

1. The text is saved into database. User gets acknowledgement.
2. The practice text will have correct number of "checked character" counted
3. Each searched character will have a correct percentage field calculated
4. Foreign key relation is established

User failed uploads text

1. The text is not saved into database. User gets error message about the fail reason.

User enters a period to load typing report

1. The overall typing record will display on the web page
2. A suggested of practice text title will show to the user

6.4 System testing

The application will be deployed to Microsoft Azure. The test will be on Azure and with both PC and mobile device to see the performance and user experience.

6.5 Acceptance testing

When I almost finish the project implementation, I will invite the first user of my application, my daughter to start using it. I will get feedback from her about user experience, or bug found by her, and then add tasks into the project backlog.

7 Methodology

7.1 Methodologies and Approach

I am going to use Agile Scrum approach to develop my typing application. The reason to choose this methodology is that Agile Scrum is good to making changes during development. The user typing experience is the most important thing in my application. I plan to let my daughter be my first test user. I am going to collect feedback from her to improve my design, so my application should be adapted to change during development. Another reason is that in the development of this application, I plan to provide the first workable version with the very basic typing functionality as soon as possible, so that I can start getting feedback from users as early as possible. If I use the traditional waterfall approach, I will design everything include all details, then after I complete the database design and creation, after the full back end feature implementation, then finish the front-end, finally I will let users to test. It is too late and possibly very difficult to modify a finished application. Therefore, Agile Scrum will let me keep testing during development and keep changing according to users' feedback.

I will add the most important and basic tasks into project backlog. I plan use two weeks as a development cycle (sprint). Before each sprint I will select tasks that I plan to finish from backlog into this sprint. During and after the sprint, I will keep adding new tasks into the backlog when necessary, because it is difficult to design every detail at the very beginning of the project. Developers tend to gradually deepen the understanding of the software during the development process. After I finish a sprint, I will go over the tasks in the backlog and rearrange them according to their levels of importance, then select what I think I can finish into the next sprint from the top of the backlog list.

By using Agile Scrum, I can easily track the progress and tasks. Even though there is no dedicated Scrum Master, I believe this approach will make my development process more organized and efficient.

7.2 Technologies Involved

This project will only use the newest and popular technologies which are also widely used in industries. All technologies used will also make sure this application is always free to use.

7.2.1 Project management

Atlassian Jira: used to track software development tasks and issues. Jira is fully compatible with Agile Scrum approach

Atlassian Bitbucket: used for version control

Sourcetree: used as a Windows client of Bitbucket

Visio 2016: used to draw design diagrams

7.2.2 Development

Visual Studio 2017: used for back-end coding, debugging and unit testing

Visual Studio Code: used for front-end coding and unit testing

Chrome: used for front-end test and debugging

Microsoft .Net Core Web API: Framework for HTTP services to communicate with client to handle CRUD operations

Microsoft Entity Framework Core 2.0: used as an object-relational mapper (ORM) to control and query database from C# code

AutoMapper: used as a C# object mapping tool

MSTest: used as back-end unit testing framework

MOQ: used as a mocking framework for C# to isolate testing class with its dependencies

Angular 7.0: used as the front-end web page development framework for both mobile and desktop

TypeScript: An Object-Oriented front-end script language which is a super set of JavaScript

Npm: this is a runtime JavaScript package manager

Karma: used for front-end unit test with Angular framework

Bootstrap 4: used for front-end UI format development

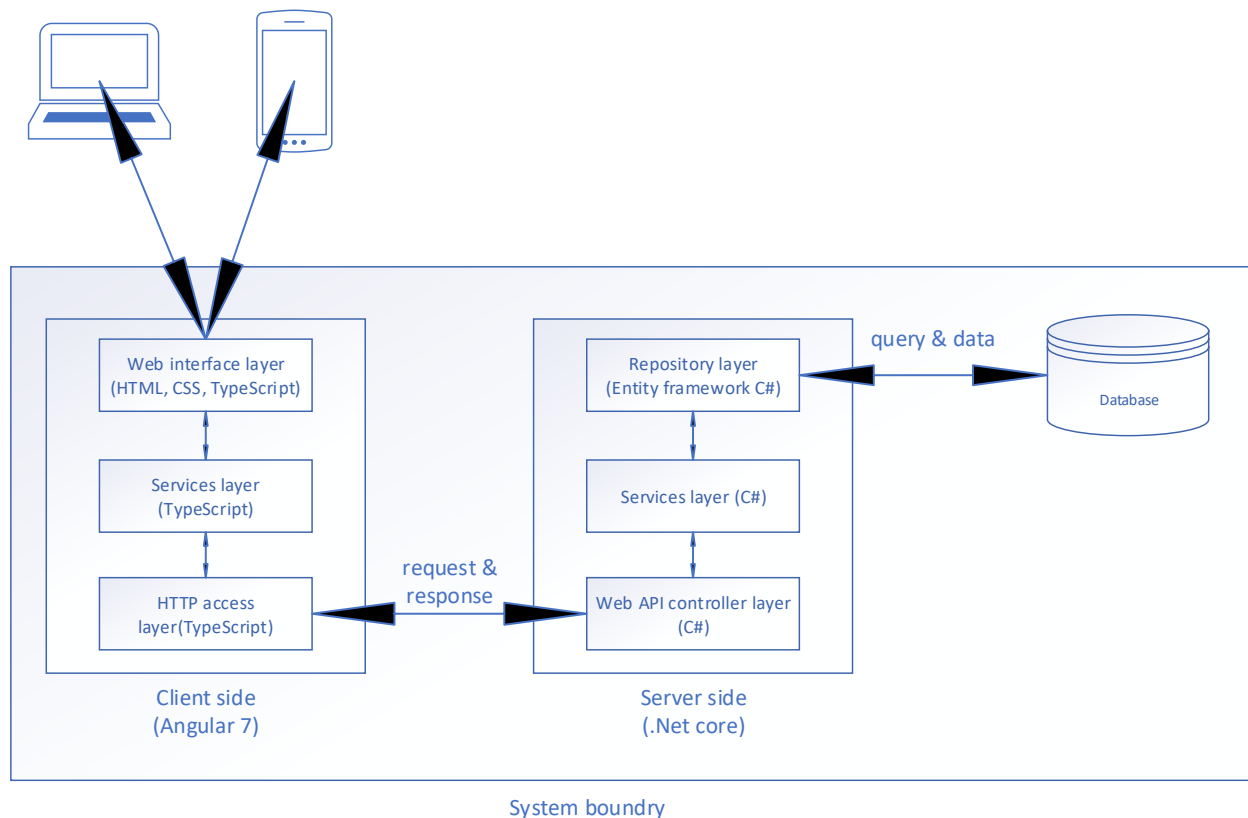
SQLite 3.26: used as a database of this application

7.2.3 Deployment

Microsoft Azure: used to host my application online

8 System/Software Architecture Diagram

The following is the system architecture diagram. It shows the overall components and the relation between them.



The overall system will have three main parts:

8.1 Client

The client side will be developed by using Google's Angular 7. All the dependency packages are managed by NPM.

Web interface layer: The entire web site pages will be divided into self-contained Angular "component". Each component is a collection of HTML, CSS and TypeScript file which contains data for its HTML page. The TypeScript file only has basic operations of data process. Angular routing will manage the relation between pages. This web interface layer will accept requests from users and display data to users. This layer is also where users practice typing.

Service layer: This layer will perform more complex data process for web interface layer and call HTTP layer to send requests to web server. These services will be injected into web interface layer when needed by using dependency injection.

HTTP access layer: This layer will prepare HTTP calling strings and call web server's web APIs.

8.2 Server

The server side will be developed by using Microsoft ASP .Net Core 2.2. It will be hosted on Microsoft Azure.

Web API controller layer: This layer acts like a gateway of the web server. It has API routing design, checks data validation and user authentication. This layer will accept the parameters from client calls and choose the right server-side services to use to request data. All server layer instances will use dependency injection to inject into controllers. When the web server responds to client requests by sending a view model of data (like a view in database to contain all data that users need for a request), this layer will decide if the returned data is valid or send back error messages.

Service layer: This layer will do two things. One, use the parameters received from controller layer and decide which repository layer's methods to call to request data, which is like data from a table. Two, the data model will be mapped to a view model to return to the controller layer. All repository instances will use dependency injection to inject into service layer.

Repository layer: This layer will use "Repository Design Pattern". Entity Framework Core is at this layer to control and query database, for example create and update database by using "code first migration". Each time the database is modified, a database migration class (C#) will be created in the project. This layer will separate the service layer logic from the actual database access. In case if we need to change database from SQLite to other databases, such as Microsoft SQL Server, we only need to modify this layer without worrying about changing the service layers.

8.3 Database

The database will use SQLite, because it's free and simple. It can also store very large amount of data up to many TB. The SQLite database file will be physically in a folder in this project. The database tables have relations by setting primary and foreign keys. It also has indices and constraints on tables. All of these are set by Entity Framework Core's code first design.

9 Innovation

First, my typing application will provide features that current typing software does not have. My design is based on my typing and language learning rationale which is to combine typing with learning vocabulary, grammar and writing skills. My typing training application will let users customize what they use to practice typing, for example, save selected text to practice. It is potentially can be used by elementary school teachers to design typing practice for students, so students can use typing to learn spelling, grammar etc. at the same time.

Second and more importantly, unlike other online typing applications only provide users very general type error reports, my typing application can suggest users what they should practice more according to their recent typing report. (The detail of the algorithm is in Technical Challenge section). Each individual learner will get different suggestions which target their specific weaknesses in typing. Teachers can save many sample essays into database, then my typing application has an algorithm to analyze the typing errors for each student and then select previously saved essays to suggest student to practice. If a

student often makes mistakes about key “w” and “o”, then either words with more “w” and “o” are suggested to this student to practice, or a pre-uploaded essay which has more “w” and “o” are suggested. This is another education rational of mine. Currently, students learn the same thing by using the same practice handed out by teachers. I believe that even though we learn the same skill and there are common learning principles, everybody’s practice path will not and should not be exactly the same, because everybody is unique in this world. Everyone has their own strengths and weaknesses, so ideally everyone’s practice should be customized based on the observation and measurement of their unique condition. Beyond the technical challenges for software development, beyond typing skill, this project is also my personal preliminary exploration about how to make education more efficient. This typing application is also a test tool for my education rational. Of course, the current technologies in our social education system are not able to make different targeted plans and exercises for each student’s learning in every subject, but I believe in the future it will become possible after internet, big data and artificial intelligence are further developed.

Let me to use a body building workout example to further explain what the innovation is in my typing application.

The following is a workout schedule for body builders.

15 pull ups

20 barbell shrugs

20 bench press

20 push ups

40 barbell squats

30 minutes cardio

Do it every Monday Thursday and Saturday.

Let’s assume some people real like this schedule and it really works for them, but will this schedule be efficient for everybody? Definitely no. That’s why a lot of bodybuilders hire a professional trainer to customize unique training programs for them based on their needs and conditions. For instance, Tom’s legs are relatively weak, so he should do more squats. Joe likes big chest, so he should do more bench press etc.

Back to our typing learning. When we open a typing software, if all learners will use all the same sample texts to learn typing, it’s like all bodybuilders use the same training program to workout. Many people hire a personal workout trainer, but who likes to hire a personal typing trainer to guide them to use the most efficient practices to increase their typing proficiency? The application I am going to develop is the personal typing trainer has the unique feature in the world for all learners and it is free. My application will measure users typing mistakes to see which keys they have higher change to type wrong.

To make learning more efficient, an important method is to minimize the time we practice on something we are already been good at and maximize the practice time on what we are not good at. Regrettably, I searched many typing training software for my daughter, currently there is no such a typing software

can act like a smart typing trainer. My online typing training application will have the innovation to fill this gap.

From technology perspective, the application I am going to develop is far from the boundary of science. The innovation selling point of my software is that currently there is no such a typing skill training software which can act like a personal typing trainer to guide each individual user through the most efficient learning path by analyzing users' typing mistakes and searching the best available typing practice texts for them. The innovation concept is not only for learning typing. I believe it is for learning anything, for example math, physics or any other topics. I believe that teachers give all students the same homework to practice is not always efficient. Each student should get different practice questions at some point of learning. If there is a software can collect all students' mistakes in their homework and tests, then analyze the root causes of why they made these mistakes and finally provide each student different practice questions which target their weak points during learning, it will be a big innovation for our society. I do not have the knowledge and ability to develop such a complex software, but I can use the same teaching and learning rational to start from a smaller task, an online typing training application. It is not a simple improvement such as making a typing software interface more attractive or something like that. My project will give the typing software an essential innovation, that is, let a typing software have the analytical ability by collecting performance data of each user and the ability to suggest users with the exercises that best enhance their learning outcomes.

10 Technical Challenges

User experience: The user interface must be intuitive and easy to use. This interface will be attractive and can immediately show users the unique features which distinguish this application from others. The front-end user interface will use some Angular animation, so it requires good understanding of Angular's animation.

Database migration with multiple schemas: The database will be managed by Microsoft Entity Framework Core by using code first methodology, so database structure will be controlled by C# code. The database in the typing application I am going to develop has multiple schemas, for example "Iam" schema has all the tables about users and roles; "Typing" schema has all the tables relate to users' typing result records. Multiple schemas can segregate the project into multiple subprojects to decouple components of the project. The project code first database migration must be able to independently control all tables in each schema, and allow tables in different schemas to reference each other, for instance, "Typing.PracticeText" table should have a foreign key to reference "Iam.User" table. However, the current version of Entity Framework Core does not support foreign key creation cross database schema. I must find a way to solve this issue.

Code maintainability and reusability: The design of the project is important. It will avoid big and long methods that handle too many tasks. Each data flow and process will be separated into difference layers, such as interface (UI on client side and API controller on server side), service (data process logic) and communication (HTTP call and data query). In each layer, all components will be designed as independent as possible, for example the user upload text UI component on client side should be able to be easily reused in other projects.

Extensibility: Developer should be able to keep adding new features to this application after the release. The code will be based on high cohesion and low coupling principle. For example, all services will use their object-oriented programming interfaces to decouple from other code, so the same logic can be easily used by new added features without code duplication.

Record analysis algorithm: This algorithm can select the best saved practice text and suggest the user to practice typing according to this user's previous type errors. It will target where the most mistakes are.

1, This typing training application only check the following characters (called **checked characters**):

- Alphabet case insensitive (26 characters)
- Number keys from 0-9 (10 characters)
- Punctuations: comma, period, single, double quote, semi colon, question mark, exclamation mark (7 characters).

Totally there are 43 keys will be checked. I only use these 43 keys as an example to explain the algorithm. In the application, I can implement a user configuration that sets which characters users want to include, such as add "<" or "%" or remove number keys etc.

2, Each time when a user uploads a new essay to the typing training application, the essay will be assigned a unique id and saved into a database table called "PracticeText". As the application checks 43 characters, a set of 43 records according to each checked character will be save to another table called "CheckedCharacter". Each record in "CheckedCharacter" table will have a foreign key reference to "Id" column in "PracticeText" table.

The "CheckedCharacter" table has the following columns:

- Id
- PracticeTextId (Foreign Key)
- Character ('a' or 'b' or other checked characters)
- **Percentage (P)** (percentage of this check character appears in this practice text, for example 1.825)

The PracticeText table has the following columns

- Id
- UserId (Foreign Key to indicate the owner of this practice text)
- Text
- TotalCharacter (how many checked characters in this text)

Each record in "PracticeText" will have 43 associated records in "CheckedCharacter" table.

Note: in the real implementation, these tables may have more columns than showed above. Here I only show the columns relate to the algorithm)

3, After a user finishes typing a practice text, a result record will be saved into database table called "TypingResult".

The "TypingResult" table has the following columns

- Id
- UserId (Foreign Key to indicate who typed it)
- PracticeTextId (Foreign Key)
- PracticeTime (include year, month day and time)

Note: users can decide if the practice this time will be save or not. If they select to save the result, then it is like a typing proficiency test for them. The result data will be used for analysis in this algorithm. If they do not want to save the result, then it means they just want to do a regular practice, not a test for future analysis.

4, Then a set of 43 detailed typing records will be saved to a table called "TypingResultDetail"

The "TypingResultDetail" table has the following columns

- Id
- TypingResultId (Foreign Key to indicate which typing result it references to)
- CharacterTypedWrong ('a' or 'b' or other checked characters)
- **Occurrence (O)** (an integer which shows how many times this checked character was typed wrong)
- **MistakeRatio (MR)** ("Occurrence" divided by "Percentage" field in "CheckedCharacter" table. Larger MR indicates user is more like need more practice on this checked character)

We need "MistakeRatio" later to calculate which checked characters this user does not type well. We use "MistakeRatio", not "Occurrence", because in a longer practice text, user have higher chance to make more mistakes. We must use the percentage of a checked character appears in a practice text as a base (O / P).

A smaller P will make the MR larger. A larger P will make the MR smaller. The reason is that if a checked character appears a lot in a practice text, then the user has more opportunities to type it wrong. For example, in the same practice text which has 1000 checked characters. Letter "e" appears 50 times, so $P(e) = 50 / 1000 = 0.05$ and letter "w" only appears 5 times, $P(w) = 5 / 1000 = 0.005$. If a user types "e" wrong 10 times, but types "w" wrong 4 times. Obviously, this user should practice more "w" rather than "e", because $MR(w) = 4 / 0.005 = 800 > 200 = 10 / 0.05 = MR(e)$.

5, Users can set how many practices results they want to keep for analyzing, for example 10 practices. After a user finishes the 11th practice, the first practice result will be deleted. The application will only keep 2nd to 11th practice results.

6. At any time, how to use a user's existing typing result to find out which practice text is the best for this user to improve typing skills? The principle is to find the practice test which has the highest checked characters appearance which this user has the highest mistake percentage.

7, Get the sum of "MistakeRatio" for each "CheckedCharacter" from the 10 saved typing result. Then sort the sum of "MistakeRatio" from high to low. For example, a user's first practice, the MistakeRatio for checked character "a" is $MR(a)_1$. The second practice MistakeRatio for checked character "a" is $MR(a)_2$. Then all the way to $MR(a)_{10}$. If the user sets to save n practice results, then the formula is

$$MR(\text{checked character})\text{sum} = \sum_{k=1}^n MR(\text{checked character})k$$

where $MR = \frac{0}{p}$ and k is saved practice results order number

8, We select the top 3 MR sums to indicate 3 checked characters which have the highest chance to type wrong by this user, for example they are “h”, “w” and “l”, so users need to practice more of these characters. In the typing training application, users can set how many highest MR they want to show, for example, they might want to know which checked characters have the top 5 MR sums. Here we use 3 MR sums as example to show the algorithm.

9, Search through the saved practiced texts which belong to this user to find practice texts with the highest percentage of “h”, “w” and “l”. The logic is as the following

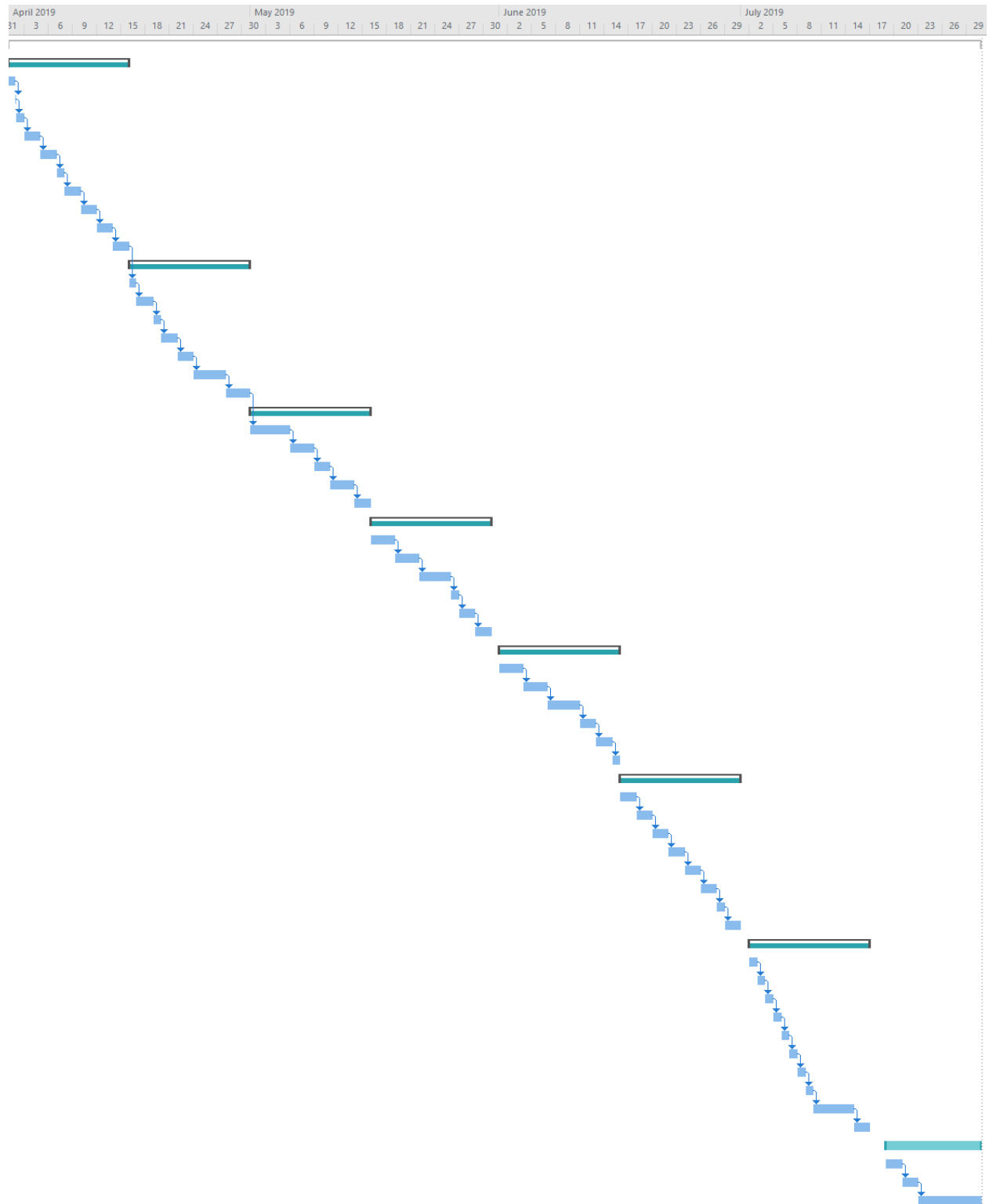
```
SELECT TOP (1) pt. Id, pt.Text
FROM CheckedCharacter cc INNER JOIN PracticeText pt ON cc.PracticeTextId = pt.Id
WHERE cc.Character = 'h' AND pt.UserId = 'currentUserId'
ORDER BY Percentage DESC
```

10, Repeat the previous logic for checked character “w” and “l”. Then suggest the 3 found practice texts to the user. It is possible that only 2 practice texts found because one text might have the highest percentage of two of the three checked characters “h”, “w” and “l”. When a text is has the highest percentage of all the three checked characters compared with other practice texts, then the user will only get one suggested practice text.

Note: This algorithm cannot prevent users from cheating. I know that type speed won’t be constant during typing, but I assume all users will do their best to type from the beginning to the end of a practice text, so the type results of how many times a character typed wrong can reflect which keys the user should practice more. However, for example, if a user knows he cannot type “b” very well, so he deliberately slows down a lot when types “b” or even look at the keyboard to avoid pressing a wrong key, then the algorithm won’t be able to suggest the best practice text.

11 Development Schedule and Milestones

Online Typing Software Development	482 hrs	Mon 4/1/19	Tue 7/30/19
▴ Sprint 1	60 hrs	Mon 4/1/19	Mon 4/15/19
Determine sprint scope	2 hrs	Mon 4/1/19	Mon 4/1/19
Create project template	2 hrs	Mon 4/1/19	Mon 4/1/19
Install & config 3rd party packages	4 hrs	Tue 4/2/19	Tue 4/2/19
Design data model	8 hrs	Wed 4/3/19	Thu 4/4/19
Implement data model for "Iam" schema	8 hrs	Fri 4/5/19	Sat 4/6/19
Data seeding	4 hrs	Sun 4/7/19	Sun 4/7/19
Web API "Iam" controller implementation	8 hrs	Mon 4/8/19	Tue 4/9/19
"Iam" service implementation	8 hrs	Wed 4/10/19	Thu 4/11/19
"Iam" repository implementation	8 hrs	Fri 4/12/19	Sat 4/13/19
Service side "Iam" test & debug	8 hrs	Sun 4/14/19	Mon 4/15/19
▴ Sprint 2	60 hrs	Tue 4/16/19	Tue 4/30/19
determine sprint scope	2 hrs	Tue 4/16/19	Tue 4/16/19
Web UI design for user login	10 hrs	Tue 4/16/19	Thu 4/18/19
Routing for user login related pages	4 hrs	Fri 4/19/19	Fri 4/19/19
Front end user login service implementation	8 hrs	Sat 4/20/19	Sun 4/21/19
User login HTTP layer implementation	8 hrs	Mon 4/22/19	Tue 4/23/19
User login and account creation related test and debug	16 hrs	Wed 4/24/19	Sat 4/27/19
Typing interface design and implementation	12 hrs	Sun 4/28/19	Tue 4/30/19
▴ Sprint 3	60 hrs	Wed 5/1/19	Wed 5/15/19
Typing logic implementation	20 hrs	Wed 5/1/19	Sun 5/5/19
Typing logic test and debug	12 hrs	Mon 5/6/19	Wed 5/8/19
Typing HTTP layer implementation	8 hrs	Thu 5/9/19	Fri 5/10/19
Back end implements data model for "Typing" schema	12 hrs	Sat 5/11/19	Mon 5/13/19
Back end Web API Typing controller implementation	8 hrs	Tue 5/14/19	Wed 5/15/19
▴ Sprint 4	60 hrs	Thu 5/16/19	Thu 5/30/19
Back end typing service implementation	12 hrs	Thu 5/16/19	Sat 5/18/19
Back end typing repository implementation	12 hrs	Sun 5/19/19	Tue 5/21/19
Back end test and debug for typing	16 hrs	Wed 5/22/19	Sat 5/25/19
User typing report data model design	4 hrs	Sun 5/26/19	Sun 5/26/19
User typing report data model implementation	8 hrs	Mon 5/27/19	Tue 5/28/19
User typing report controller implementation	8 hrs	Wed 5/29/19	Thu 5/30/19
▴ Sprint 5	60 hrs	Sat 6/1/19	Sat 6/15/19
User typing report service implementation	12 hrs	Sat 6/1/19	Mon 6/3/19
User typing report repository implementation	12 hrs	Tue 6/4/19	Thu 6/6/19
Back end user typing report test and debug	16 hrs	Fri 6/7/19	Mon 6/10/19
Front end typing report request service implementation	8 hrs	Tue 6/11/19	Wed 6/12/19
Front end typing report request HTTP layer implementation	8 hrs	Thu 6/13/19	Fri 6/14/19
Front end report request test	4 hrs	Sat 6/15/19	Sat 6/15/19
▴ Sprint 6	60 hrs	Sun 6/16/19	Sun 6/30/19
Front end file upload UI implementation	8 hrs	Sun 6/16/19	Mon 6/17/19
Front end file upload service implementation	8 hrs	Tue 6/18/19	Wed 6/19/19
Front end file upload HTTP layer implementation	8 hrs	Thu 6/20/19	Fri 6/21/19
Back end file upload data model implementation	8 hrs	Sat 6/22/19	Sun 6/23/19
Back end file upload Web API controller implementation	8 hrs	Mon 6/24/19	Tue 6/25/19
Back end file upload service implementation	8 hrs	Wed 6/26/19	Thu 6/27/19
Back end file upload repository implementation	4 hrs	Fri 6/28/19	Fri 6/28/19
Back end file update test and debug	8 hrs	Sat 6/29/19	Sun 6/30/19
▴ Sprint 7	60 hrs	Tue 7/2/19	Tue 7/16/19
Front end file loading UI implementation	4 hrs	Tue 7/2/19	Tue 7/2/19
Front end file loading service implementation	4 hrs	Wed 7/3/19	Wed 7/3/19
Front end file loading HTTP layer implementation	4 hrs	Thu 7/4/19	Thu 7/4/19
Back end file loading Web API controller implementation	4 hrs	Fri 7/5/19	Fri 7/5/19
Back end file loading service implementation	4 hrs	Sat 7/6/19	Sat 7/6/19
Back end file loading repository implementation	4 hrs	Sun 7/7/19	Sun 7/7/19
Back end file update test and debug	4 hrs	Mon 7/8/19	Mon 7/8/19
File loading test and debug	4 hrs	Tue 7/9/19	Tue 7/9/19
Typing error analysis algorithm implementation	20 hrs	Wed 7/10/19	Sun 7/14/19
System integration test	8 hrs	Mon 7/15/19	Tue 7/16/19
Sprint 8	46 hrs	Fri 7/19/19	Tue 7/30/19
Deployment to Azure	8 hrs	Fri 7/19/19	Sat 7/20/19
User manual	8 hrs	Sun 7/21/19	Mon 7/22/19
Final Report	30 hrs	Tue 7/23/19	Tue 7/30/19



The project schedule is written in Microsoft Project. I must separate the entire project into 2 pieces from middle to fit the proposal page width.

I plan to start implementing the project from March 1st and finish it before the end of June. There will be 8 development cycles (sprint). Each sprint has around 60 hours. I plan to use 462 hours to finish this project.

12 Deliverables

The deliverables of this project are as follows:

- Web server in ASP .Net Core
- Web site in Angular TypeScript and JavaScript
- Sample database in SQLite
- Final Report: final report of this practicum course
- User Manual: user manual to show how to user this application

13 Conclusion and Expertise Development

This project will help me to improve my software developing skills from database, web server to front-end web site. This project will also include all the software planning, design and management skills I have learned from BCIT and the software development team in Schneider Electric. There might be some loopholes in my learning from school and work, but in this project, I will find out and make up for these loopholes, so that I can become a better software developer.

14 References

- Angular framework <https://angular.io/>
- TypeScript <https://www.typescriptlang.org/>
- Bootstrap <https://getbootstrap.com/>
- Node.js <https://nodejs.org/en/>
- NPM <https://www.npmjs.com/>
- ASP.NET Core <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>
- Entity Framework Core <https://docs.microsoft.com/en-us/ef/core/>
- AutoMapper <https://automapper.org/>
- Moq <https://github.com/moq/moq4>
- SQLite <https://www.sqlite.org/index.html>
- Azure <https://azure.microsoft.com/en-ca/>
- Jira <https://www.atlassian.com/software/jira>
- Bitbucket <https://bitbucket.org/product>
- Sourcetree <https://www.sourcetreeapp.com/>

15 Change Log

Version 2 on Feb 23rd, 2019

- Cover page, change the project name from “Online Typing Application” to “Online Typing **Training** Application” to emphasize this software is for users to improve typing skills. It’s not a software for typing essays like Microsoft Word. So called essays are just practice text.
- Section 3 “Problem Statement and Background”, addition more explanations about what exactly the problems my project will solve. Clear the misunderstanding about grammar checking. Page 5-7.
- Section 9 “Innovation”, additional explanations about the advanced learning / training rational of my project and why it’s innovative and unique to solve problems which not yet have a solution in the world. Remove a paragraph that mentions about using cutting age technologies. Page 15-16.
- Section 10 “Technical Challenges”, add algorithm detail. Page 17-19.
- Section 11 “Development Schedule and Milestones”, update schedule. Page 20-21.