Digital Language Training System

Software Requirements Specification

<Version 1.5 >

<10-21-2019>

Tanay Ghosh

Todd Hricik

Reneel Pamarthi

Prepared for: Dr. Samir Padalkar

CS 683 Project Management

Fall 2019

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Author | Comments |
| 09/23/2019 | 1 | Tanay  Todd  Reneel | Initial Version |
| 09/30/2019 | 2 | Tanay  Todd  Reneel | Added –  Functional requirements  Non-functional requirements  Activity/Data flow diagrams |
| 10/07/19 | 3 | Tanay  Todd  Reneel | Added-  System Architecture Diagram and Summary  Use Cases  POS edits |
| 10/14/19 | 4 | Tanay Todd Reneel | Added –  Functional requirements Work Breakdown Structure  Association with RBS |
| 10/21/19 | 5 | Tanay  Todd  Reneel | Added WBS to RBS  Team Definitions  Time/Resource Estimates |

Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| Signature | Printed Name | Title | Date |
|  |  |  |  |

Table of Contents

1. Overview
2. Project Goal and Solution
3. Functional Requirements
4. Non-functional Requirements
5. System Architecture
6. Use Cases
7. Algorithms
8. [References 26](#_Toc21349652)

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Overview Statement** | Project Name:  Digital Language Training System | Team: Red |  |
| Problem/Opportunity: Our world is becoming ever increasingly connected across social and business landscapes as a result of internet-based technologies that can bring together individuals from disparate communities which have traditionally been localized according to geographic constraints. Today individuals and business firms are able to connect and interact within a radius of coverage that covers most of the developed world. As a result, communities have become more diverse as people coming from all cultures and locations can more easily connect, interact and transact with one another. While geographical boundaries have been broken via such technologies, there still exists language barriers amongst the many connected individuals who are unable to speak and understand the native tongues of those to whom they are connected.  The online language arts learning market is expected to be between $4-$8 billion during by 2024 [1] [2]. Indeed there is a recognizable demand for language training products worldwide who’s market majority is currently shared between two firms: Rosetta Stone Ltd. [3] and Duolingo Inc. [4] [2]. Rosetta Stone Ltd. offers a subscription based digital training platform that covers 30 languages through an online classroom styled delivery and also offers live one-on-one online tutoring via virtual web meetings [5] and has developed a social network presence within which users can interact on Facebook [6] and Twitter [7]. Duolingo Inc. covers 26 languages (including Klingon for avid Star Trek Fans) using a gamification styled delivery which is based on both an ad-free and fee based subscription models and provides an integrated social community that allows users to follow and compete with others in their Duolingo network [4].  Although the market for digital language arts training applications have been reached by several competing entities, we view opportunities to be captured within the space. Our analysis of the products offered by Rosetta Stone and Duolingo are disjoint along both the personal tutoring and gamification components. In our view there is room for significant improvement to the level of interactivity and engagement possible through a digital language art learning platform. Namely we recognize the potential for value to be added by offering, in addition to language arts training via traditional modalities offered in existing solutions, a cognitive computing based solution that creates an intuitive virtual tutor than can be available to the user on demand via an audio-visual-text-social media based interface. Our aim is to 1.) hybridize traditional digital language arts learning modalities having demonstrated market success, 2.) expand upon current platform technologies offered using cognitive computing based solutions, and 3.) integrate said technologies into a uniquely defined product that offers market leading high quality and engaging language arts training that is cognizant of each individual user. | | | |
| Goal: Provide a multi-platform digital language training system that engages the user through a cognitively aware gamified interface. | | | |
| Objectives:   1. Provide a language arts training regimen via an interactive digital interface which implements the following modalities 2. Support for text and pictographic based multiple-choice exercises 3. Support for text and pictographic based fill in the blank exercises 4. Support for “listen” and “respond” based exercises where user responses may be speech or written text 5. Maintain user engagement via the following modalities 6. Support for modular learning where modules are used a micro-course focusing on one aspect of the language learning process 7. Support for beginner, intermediate and advanced levels of difficulty for each language learning task 8. Support for gamified based user performance-based reward system that considers the level of difficulty, consistency, frequency of and performance of the user’s engagement with the learning system exercises 9. Support for in app community connectivity allowing for users to connect, socially interact and compete in user group created learning competitions 10. Support for an individualized virtual cognizant tutor and learning companion 11. Support for user to set and adjust weekly goals for performance and engagement and continuously have access to feedback metrics related to user goals. | | | |
| Success Criteria:   1. Aggregate and stratified user performance according to goals set by the user (See Objectives: section D) being met and exceeded. We will consider our product to be successful if users consider themselves to be successful during their learning journey. We aim to have at least 90% of our users reach at least 90% of the goals they set. 2. Size of user base compared to current market leaders 3. User learning system usage metrics from (aggregate and stratified) based upon 4. Frequency of weekly usage on daily scale > 5 days per week 5. Number of lessons attempted vs completed across all users > 90% 6. Frequency of exercises receiving correct response within the first attempt > 80% 7. User progression metrics that track for language skills development within and among beginner, intermediate and advanced levels: We aim to see all users progress through all modules of each level of difficulty consistently according their achievement of the short- term goals they set weekly. | | | |
| Assumptions:   1. The development and QA team are well equipped to build and deliver the software that can run even on the low-end devices with relatively slow internet speed. 2. There is a bug reporting system in place where users can report issues that they run into while using the application. These bug reports need to be read, analyzed and assigned to the correct team so that the issues can be fixed in a timely manner. 3. There will exist teams of composed of senior management, product/project managers, language experts, gamification experts, and social media designers who will work within cross-functional groups including all relevant UI, machine learning and database developers throughout the adaptive agile development and launch processes. | | | |
| Risks:   1. All requirements not being identified at the beginning of the development phase leading to a requirement inflation at the later stages of the project and could threaten the budget estimates and deadlines. 2. The system might be prone to hacking and can lead to theft of user information and loss of data. 3. Server breakdown could lead to loss of data if there are no backup servers in place. | | | |
| Obstacles:   1. Designing the application in such a manner that the reading, writing and speaking skills of the user learning a new language is fully tested. 2. Designing the translation algorithm such that it presents the most appropriate translation to words and phrases depending on the context. 3. Monitoring and analyzing the usage metrics and figure out if user activity has dropped and coming up with new features to make the application more engaging for the users. 4. Designing the system in such a manner that it can be used offline. 5. Making the system platform independent so that the user experience is uniform regardless of the device the user is using to run the application. | | | |
| Prepared By:  Team Red | Date: | Approved By: | Date: |

1. Overview

The purpose of this document is to present the pre-planning related project components that composes the software requirements (functional and non-functional), use cases, system architecture and hardware requirements of the language learning system. This document will compose a detailed description of the planning phase of the language learning system development process and includes details of how cross-functional teams will be organized and which deliverables each will be responsible for. This document analyzes and quantifies the factors/events that may exist at any point during the planning and launch processes.

1. Definitions

3. Functional Requirements

3.0 Application launch and entry point

3.0.1 Splash screen will appear upon application launch (Diagram 1)

A picture containing object

Description automatically generated

Diagram 1: Splash Screen

3.0.2 A new user will be presented with the New User Account Setup Interface

3.0.3 An existing user will be presented with the training program interface (Diagram 2)(See 3.4)

A screenshot of a cell phone

Description automatically generated

Diagram 2

3.0.3.1 Lower button bar of main interface will provide access buttons to

3.0.3.1.1 User training program interface

3.0.3.1.2 User social media tools interface

3.0.3.1.3 Weekly League Standings interface

3.0.3.1.4 App Store interface

3.0.3.1.5 The identity of the interface in which the user currently resides will be

indicated by the coloring of the appropriate lower button bar access button. All other lower button bar access buttons will be colored grey while User remains in the current interface:

Case 1) User is in Training Program Interface (Purple)

Case 2) User is in Social Media Interface (Orange)

Case 3) User is in Weekly League Standings Interface (Yellow)

Case 4) User is in App Store Interface (Black)

3.0.3.2 Upper button bar of main interface will provide

3.0.3.1.1 Language selector access button

3.0.3.1.2 Button to access Health and Practice interface

3.0.3.1.3 Button to access Trophy Collection interface

3.0.1.1.3 Graphical icon displaying number of Rewards Points earned

3.0.3.3 The area between the Upper and Lower Button Bars will be referred to as the Main Window (Diagram 3)

A screenshot of a cell phone

Description automatically generated

Diagram 3

3.0.3.4 Upon first entry to the Language Training Application, User will be allowed to navigate among Interfaces as shown in (Diagram global\_interface\_navigation\_paths)

A close up of a map

Description automatically generated

Diagam global\_interface\_navigation\_paths

3.1 Language(s) Selection Interface:

3.1.1 This interface will present the collection of all available language Training Programs

available within a scrollable action window

3.1.2 Each language Training Program will be represented by a Training Program selector button

that contains a picture of the corresponding country’s flag.

3.1.3 Selection of a language Training Program via tapping on a given Training Program selector button within the Language Selection interface will bring User to the Training Program interface for the selected language

3.2 Training Program Interface

3.2.1 Training Program Conceptual Organization

3.2.1.1 A Training program will consist of levels

3.2.1.2 Each consecutive level will represent a higher degree of learning difficulty.

3.2.1.3 Training program levels will consist of modules.

3.2.1.4 Each module will contain a set of lecture notes related to the module topic.

3.2.1.4 Each Module will consist of a collection of Parts

3.2.1.5 Each Part will consist of a collection of Lessons

3.2.1.6 Each Lesson will consist of a collection of interactive Exercises related to the Module topic.

3.2.2 Training program progression concepts

3.2.2.1 Progression from one training level to the next more difficult level will be allowed once all modules at the lower level(s) of difficulty have been completed satisfactorily.

3.2.2.2 Satisfactory completion of any module is defined as satisfactory completion of all lessons within the module.

3.2.2.3 Satisfactory completion of a lesson is defined as user having provided correct responses to all exercises within a single lesson attempt.

3.2.3 Graphical Representation of a Training Program (Diagram 4)

A screenshot of a cell phone

Description automatically generated

Diagram 4

3.2.3.1 A Training Program will be represented as a collection of mutually exclusive directed acyclic graphs where each graph will represent a recommended learning path for each Level of difficulty.

3.2.3.2 Each module will be represented as a node in a graph having an in degree of one and an out degree of one.

3.2.3.3 Recommended learning path will be represented as edges connecting each consecutively recommended module (node) within a graph for a given level.

3.2.4 Training Program Interface Functions and Behaviors: The Training Program Interface will

present all Levels of a Training Program within a scrollable view of the Main Window.

3.2.4.1 Completed module nodes will be marked as completed graphically via a “check mark” graphic place in the upper right-hand corner of each completed module.

3.2.2.2 Edges between a pair of two completed modules will be graphically distinct from edges between all other possible module pairs.

3.2.4.3 User may select any training module within their current Training Program Level by tapping on a node within the accessible graph. After tapping a module (node) a pop up window will provide the following (Diagram selected module pop\_up\_window)

A screenshot of a cell phone

Description automatically generated

Diagram selected\_module\_pop\_up\_window

3.2.4.3.1 Textual information informing the user of the Lesson Part and Lesson Number which is about to begin

3.2.4.3.2 An access button to the Introductory Notes Interface applicable for the current lesson.

3.2.4.3.3 An access button that will trigger the Current Lesson Attempt Interface to begin.

3.2.4.3.4 See Section 3.4.5 for detailed Lesson Functional Requirements.

3.2.5 Training Program Lesson:

3.2.5.1 Lesson Introductory Notes Interface

3.2.5.2 Lesson Attempt Interface: Upon triggering a Current Lesson Attempt to begin, User will be presented with a set of Lesson Attempt Global Components and a sequence of interactive Exercises.

3.2.5.2.1 Lesson Attempt Interface Global Components: During any point of a Lesson, the User will be presented with the following elements (Diagram lesson\_attempt\_global\_elements).

A screenshot of a cell phone

Description automatically generated

Diagram lesson\_attempt\_global\_elements

3.2.5.2.1.1 Lesson Exit Button

3.2.5.2.1.2 Lesson Progress Bar

3.2.5.2.1.3 Life Status Meter: Displays the number of Health Points the User has remaining.

3.2.5.2.1.4 Interactive Exercise

3.2.5.2.2 Interactive Exercises: Each Exercise Interface will be structured according to one of the following Exercise Types

3.2.5.2.2.1 Textual Challenge/Textual Response: User will be asked to read a Challenge Sentence and provide the correct translation via the keyboard. The Challenge Sentence may be written in either the Training Program Language or the user’s native language

3.2.5.2.2.1.1 User will be notified as to whether their response is correct or incorrect as shown in (Diagram exercise\_textual\_challenge\_textual\_response\_results).

A close up of a device

Description automatically generated

Diagram exercise\_textual\_challenge\_textual\_response\_results

3.2.5.2.2.2 Textual Challenge/Multiple Choice Response: User will be asked to read a Challenge Sentence and several potentially correct textual translations, each embedded within a button. User will select their guess as to the correct translation by tapping the button containing the text representing the correct translation. The Challenge Sentence may be written in either the Training Program Language or the user’s native language.

3.2.5.2.2.2.1 User will tap the translation button which they believe is correct which will be highlighted.

3.2.5.2.2.2.2 User will click NEXT button and will then be notified as to whether their response is correct or incorrect as shown in (Diagram exercise\_textual\_challenge\_mult\_choice \_results). Correct answer will be highlighted.

A screenshot of a cellphone

Description automatically generated

Diagram exercise\_textual\_challenge\_mult\_choice \_results

3.2.5.2.2.3 Audio Challenge/Textual Response: User will be asked to listen to an audio clip of a phrase spoken in the Training Program Language. User will then enter their translation via the keyboard.

3.2.5.2.2.3.1 Audio recording may be played by User via tapping an Audio Challenge Button which activates the audio clip based Challenge Sentence.

3.2.5.2.2.3.2 Audio clip may be played repeatedly via tapping the Audio Challenge Button until User inputs their response and submits their answer.

3.2.5.2.2.4 Textual Challenge/Audio Response: User will be asked to read a textual Challenge Sentence and provide the correct translation via the microphone (Diagram textual\_challenge\_voice\_response).

A screenshot of a cell phone

Description automatically generated

Diagram exercise\_textual\_challenge\_mult\_choice \_results

3.2.5.2.2.4.1 User will tap the Microphone Button when prepared to record their translation. User will speak their translation into the microphone and tap the Microphone Button again to finish recording. If recording is verified to contain machine translatable speech, a SUBMIT button will be activated and User may proceed to receive results. If recording is not verified to contain machine translatable speech, USER will be brought back to initial Exercise state so that User may repeat the recording process.

3.2.5.2.2.5 Audio Challenge/Audio Response: User will be asked to listen to an audio clip of a phrase spoken in the Training Program Language. User will then enter their translation by speaking their translation into the microphone. Requirements 3.2.5.2.2.3.1, 3.2.5.2.2.3.2 and 3.2.5.2.2.4.1 apply.

3.2.5.3.2 Exercise Tools: Once User has submitted their response and received the results for a particular exercise, two Exercise Tools Buttons allowing User to access the Cognizant Language Tutor and or flag the exercise to the application maintainers for potential problems (Diagram Exercise Tools).

3.2.5.3.2.1 Cognizant Language Tutor

3.2.5.3.2.1.1 USER can tap the Cognizant Language Tutor button to flag an exercise as being difficult or worth revisiting later as a review.

A screenshot of a cell phone

Description automatically generated3.2.5.3.2.2 Flag for Problematic Exercises Button will bring up a popup window having a text container within which User may provide textual details related to the Exercise and a Post Flag button which will submit the Flag to the system database.

Diagram exercise\_tools

3.3 Trophy Collection Interface This interface will provide a collection of Trophy icons that the user may earn while using the app.

3.3.1 Description of Trophies which can be earned

3.3.1.1 Accuracy Master: An Accuracy Master trophy will be awarded when User

completes a given number of Lessons without getting any Exercises wrong.

3.3.1.1.1 Level I: Complete one lesson without getting any Exercises wrong

3.3.1.1.2 Level II: Complete five Lessons consecutively without getting

any Exercises wrong.

3.3.1.1.3 Level III: Complete twenty Lessons consecutively without getting

any Exercises wrong.

3.3.1.2 Consistency Master: A Consistency Master trophy will be awarded when User

satisfactorily completes at least one Lesson each day over the course of a number of

consecutive days.

3.3.1.2.1: Level I: Seven consecutive days

3.3.1.2.2: Level II: Fourteen consecutive days

3.3.1.2.3: Level III: Thirty consecutive days

3.3.1.3 Weekend Warrior: A Weekend Warrior Trophy will be awarded when User

completes at least one lesson each day over the course of a consecutive Friday,

Saturday and Sunday weekend.

3.3.1.4 Night Hawk: A Night Hawk trophy will be awarded when User satisfactorily completes a Lesson between the hours of 10:00 p.m. and 12:00 a.m.

3.3.1.5 Morning Glory: A Morning Glory trophy will be awarded when User satisfactorily

completes a Lesson between the hours of 5:00 a.m. and 9:00 a.m.

3.3.1.6 Social Butterfly: A Social Butterfly trophy will be awarded when User connects

with a given number of other users via the in-app Social Network Interface

3.3.1.6.1 Level I: User connects with three new friends

3.3.1.6.2 Level II: User connects with five new friends

3.3.1.6.3 Level III: User connects with ten new friends

3.3.1.7 High Scoring Star: A High Scoring Star trophy will be awarded when….

3.4 League Standing Interface: Weekly league competitions will be held every week. Each league competition will begin on Sunday at 12:00 a.m. GTC and end on 11:59 p.m. UTC the following Sunday.

3.4.1 There will exist a finite number of leagues. All new USERs will automatically start in League 1. The top 10% of league members having the most XP points at the end of the week will be promoted to the next league up for the following week’s competition. The bottom 10% of each league will be demoted to the league below for the following week’s competition. All other users will remain in their current league for the next week’s competition.

3.4 League standings for USER’s League will be presented in the action window of the League Standing Interface. There a scrolling list of all other USERs within the USER league will be provided along with the number of XP points each has accumulated in the current week’s competition. USERS in top and bottom 10th percentiles will be separated within the list by a cutoff line demarking those who would be promoted to the next higher league (and those who would be demoted to the next lower league) if the week’s competition were to end today. This standings list will be updated in real-time as each USER completes a lesson and thus triggers their weekly accumulated XP point score to be updated.

3.5 Health and Practice Interface: Each new USER will be given three health points. One health point will be deducted for each Exercise USER answers incorrectly.

3.5.1 Once USER’s health points reach zero, their progress in the Language Training Program is reset to the initial state of a new USER.

3.5.2 USER may increase their health score by completing a block of Practice Exercises. One additional health point will be given for each block of exercises completed while answering all Practice Exercises correctly within the block.

3.6 Weekly XP Points tracking: Beginning on Each Sunday at 12:00 a.m. GTC, a weekly XP point tally will be

set to zero and a running tally of the XP points accumulated by the user as lessons are completed.

3.6.1 Each lesson will allow for the opportunity to earn a maximum of 15 XP points by answering

all exercises within the lesson correctly. The User will begin the lesson having all 15 XP points and will lose one XP point for every exercise answered incorrectly during the lesson. Completion of a lesson will trigger the XP points earned during that lesson to be added to their weekly total.

3.7 Rewards Points: After a lesson is completed, USER will be presented with a “Pick A Card” game that allows them to earn “Rewards Points”. Three Cards will be presented to the USER face down. Each Card will have a number on the bottom side which constitutes the number of Rewards Points to be awarded. User will select one of the three Cards and the Card’s Rewards Points Award will be revealed to the USER. The USER will then be given the option to watch an advertisement. If the USER opts to watch the advertisement, the USER will be permitted to select another Card from the remaining face down Cards in order to receive more Rewards Points.

3.8 Advertisements: After the Rewards Points game has been played, The USER will be given a An advertisement will be played at the end of each completed lesson.

3.9 App Store Interface: The App Store interface will present the USER with special privileges that can be received by trading in each amount of Rewards Points.

3.9.1 Privilege 1: Restore Health Points to full health (3 Health Points)

3.9.2 Privilege 2: Move into the next higher league

3.9.3 Privilege 3: Unlock the next higher Training Program Level without having finished the current Training Program Level.

3.9.4 Privilege 4: Make a “Double or Nothing” bet on the XP points earned in the next Lesson taken and completed.

3.9.4.1 Purchasing this privilege will result in the USER earning 2X the XP points earned in the following lesson attempt.

3.9.4.2 Failing to complete the next lesson results in the USER’s bet being forfeited.

3.9.4.3 Completing the first lesson after placing the “Double or Nothing” bet will result in the XP points being awarded in double the magnitude.

3.9.5 Privilege 5: Set a weekly Goal: This privilege will allow the user to set a Weekly XP points goal. User will be rewarded 3X XP points for that week if the goal is met.

3.9.6 Privilege 6: Unlock Bonus Learning Modules not included within the Language Learning Program.

3.9.6.1 These modules will be outside of the scope of the Language Learning Module and will provide the USER with an opportunity to learn more language concepts in addition to those included within the Language Learning Module. Opportunities to exchange Rewards Points for unlocking bonus modules will be presented during each Level of the Language Training Program.

3.9.6.2 The requirements of these bonus modules will be the same as those requirements listed for the modules which are included in the Language Learning Program.

3.10 Cognizant Tutor: Each new USER will be associated with a personalized digital tutor/mentor that will monitor the USER’s progression through each level of a given Language Training Program and look for associations between the USER’s application usage statistics and the Language Training Program Exercises data.

3.10.1 The Cognizant Tutor will periodically send messages to the USER to let them know that an association rule had been found. The Cognizant Tutor will report the association using non-technical language and describe reasons why that association rule may have been found. For example, if the USER answers many more Exercises incorrectly during late night hours, the Cognizant Tutor will message the USER saying “Hi USER! I noticed that you answer many more Exercises correctly during the daytime hours. It shows great effort to train during the night time hours, but if you are looking to get promoted to the next league, you might score better on your lessons if you do them during the day.”

4. Non-functional requirements

4.1 Performance: The UX should be intuitive and easy to use. The application should be fast and not put

too much pressure on the hardware.

4.1.1 Battery: The app should be able to run on background consuming the least amount of

power that is possible. User should be able to run the application from the background and

should be able to seamlessly resume using the application.

4.1.2 Network Connectivity: The app should work on poor network connectivity. If the network is

extremely poor and the app fails to run then the user should be notified with a meaningful

notification message. The app should resume seamlessly when the network improves.

4.1.3 Speed: The app should run fast and smoothly. The lectures should not take more than

700ms to 1s. The app should navigate from one page without lagging and the transition

Animation should not appear jumpy.

4.1.4 Load balancing: The server(s) should be able to handle large volume of requests and should

be able to balance the load in case of heavy traffic.

4.2 Scalability

4.2.1 The architecture of the system needs to be designed in such a way that it is scalable. It

should be easy for course creators to add new courses.

4.2.2 The frontend needs to be designed in such a manner that new added courses are displayed

without any UI distortion. Courses can be of different structure, so, the frontend should be able

to handle different types of courses with little customization.

4.2.3 The backend also needs to be designed in such a way that it allows customization.

4.3 Reliability

4.3.1 The app should be available for use 24 hours, every day. In case of technical faults, there

should be backups so there is no down time.

4.3.2 Our application offers offline support, so, there should be no difference between the online

experience and the offline experience. There shouldn’t be any continuity issue.

4.4 Usability

4.4.1 There should be a brief tutorial for new users to learn the important aspects of the

application and get acclimatized using it.

4.4.2 Users should be allowed to use Facebook and Google accounts to register for the

application.

4.4.3 The courses must get more challenging as the user’s performance gets better.

4.5 Portability

4.5.1 The application should be platform independent i.e. should run on android, iOS and web

browsers.

4.5.2 There should not be any difference in the user experience while using the application on

any of the fields. The only acceptable difference is that the UI needs to adhere with the frontend

standards that applications need to follow for these individual platforms.

4.6 Security

4.6.1 The users can create account using email and password or can create account using

Facebook or Google accounts.

4.6.2 Users allowed to use Facebook and Google accounts to login along with the email and

password combination.

4.6.3 The password should adhere to the following rules:

4.6.3.1 Should be alphanumeric

4.6.3.2 Must have at least one Upper case letter and one lower case letter

4.6.3.3 Must have a special character

4.6.4 Users should be prompted to change the password every 3 months

4.6.5 Sensitive information like name, date of birth, payment info, phone number and address

should be encrypted before being saved in the database.

4.6.6 Email should be verified using verification link and phone number needs to be verified by

send in security code to the phone number.

4.6.7 All requests need to be encrypted using an SSL certificate and the requests needs to be

HTTPS requests.

4.6.8 The system should logout a user every time the password is changed to avoid unauthorized

access.

4.6.9 The user should be able to easily recover or reset password in case the password is

forgotten.

4.6.10 The security protocol needs to implement uniformly throughout the system for all

servers and backup servers.

4.7 Privacy

4.7.1 Under no circumstances is the user information shared with anyone.

4.7.2 Password field in the frontend is masked with an option to “Show password”. Password is

encrypted first and then saved in the backend.

4.7.3 Payment info is partially masked while displaying on the frontend and encrypted before

being saved in the backend.

4.7.4 Password can only be reset by the user. The user can request a system administrator to

reset the password.

4.8 Reusability

4.8.1 Content creators can use existing course template to create new contents or can create a

new template. The system should allow that level of customization.

5. System Architecture

The language training system will implement the MVC model wherein the user will interact with the system through a mobile app front end that provides application Views. All application Views will be presented to the user according to the Presentation Logic which provides the implementation of all methods and constraints related to delivering any requested application View. Within the Middle Layer, a Controller will accept message requests from the current application View and call the appropriate local and/or remote data using the appropriate services and pass the completed request back to the User via the application View. Workflows will include the application interface navigation constraints found in functional requirement 3.0.3.4 and the methods constraining the training progression concepts (functional requirement 3.2.2). All methods connecting the training program interface (functional requirements 3.2.4-3.5) to user data and services will be defined in Components and specific instances of those Components will be stored as Entities on the User device. Local Data will consist of all User specific data related to functional requirements 3.2-3.5. Local Data Accessors will interface among Controller requests, Local Data and any Remote Data requested by local Data Services and returned by Remote Data Access functions. Remote Data Services will provide methods for the selection and aggregation of the requested. Data Utilities will ensure that all appropriate Data Service methods are properly constrained. Common application development-based configurations including necessary libraries, global application parameterizations and any other application dependencies will be provided by Configuration. Additionally, Security will provide all security policies and security functionalities required in non-functional requirements. Means of all communications between the User via the Front End Views, the Middle Layer and any local/remote data will be contained within communications including all audio codecs used during lessons and/or cognizant tutoring sessions, all network related infrastructure functions/configurations required for accessing Remote Server Data [8].

A screenshot of a cell phone

Description automatically generated

6. Use Cases

6.1 User management use cases

|  |
| --- |
| **Action:** New user registration |
| **Brief Description:** A new user registers for the application. |
| **Actors:** New user |
| **Pre-conditions:** The customer has downloaded the application for play store/ app store and launches the application. The user also has respectable network connectivity on the device. |
| **Basic Flow of events:**   1. The user is given the option to register either using Google and Facebook account or by entering Name, date of birth, phone (optional), email address and password. 2. If the user chooses to register using either Google or Facebook account, the application registers the user through the social networking platform selected by the user. The user lands on the language selection page. 3. If the user chooses to register by entering all the details, the user is sent an email verification link to the entered email address. Once the user verifies the email address using the verification link, the user lands on the language selection page. The user will not be able to move ahead with the application unless the email address is verified. If the user launches the application without verifying the email address, he will be shown a message saying, “Please verify the email address to use the application”. |
| **Extensions:**   1. An existing user launches the application for the first time.    1. The user can choose to login instead of register.    2. The user lands on the login page. |
| **Post Conditions:** |
| **Special Requirements:** The user has access to network and a browser, an android or an iOS phone. |

|  |
| --- |
| **Action:** Existing user login |
| **Brief Description:** An existing user logs in to the application. |
| **Actors:** Existing user |
| **Pre-conditions:** The user is registered and has network connectivity. |
| **Basic Flow of events:**   1. The user can login either using Facebook and Google accounts or by entering the email ID and password. 2. If the user chooses to login using either Facebook or Google account, he can click on the respective buttons and login to the application through the corresponding social networking account. The user lands on the language selection page. 3. If the user chooses to login using email ID and password, the user is logged in after verifying that the email ID and password entered by the user is correct. The user lands on the language selection page. |
| **Extensions:**   1. The user can select “Keep me signed in” during login. If the user chooses that option, the email ID and password is encrypted and saved on the local device. This info is used to auto-login the user. |
| **Post Conditions:** |
| **Special Requirements:** |

|  |
| --- |
| **Action:** Forgot password |
| **Brief Description:** If the user forgets the password, he can reset it. |
| **Actors:** Registered user |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. The user clicks on the “Forgot password” link. 2. The user is prompted to enter the registered email ID. 3. If the entered email ID exists in the system,    1. A password reset link is sent to the email ID.    2. The user clicks on the link and is redirected to enter the new password.    3. User enter the password which adheres to our rules.    4. User is redirected to the login page. 4. If the entered email ID does not exist in our system, the user is shown an info message saying “The email ID entered does not exist in our system. Please enter the correct email ID or click here to register.” The “click here to register” in the message above would be a link which redirects the user to the registration page. |
| **Extensions:** |
| **Post Conditions:**   1. The user can login using the new password. 2. The old password will NOT let the user login. |
| **Special Requirements:** |

6.2 Training program interface use cases

|  |
| --- |
| **Action:** Language selection |
| **Brief Description:** The user can select languages that he is interested to learn |
| **Actors:** Registered user |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. The user lands on the language selection page and sees the languages offered by the application. 2. The user selects the language that he is interested to learn. 3. The user lands on the training program page for that language. |
| **Extensions:** |
| **Post Conditions:** |
| **Special Requirements:** |

|  |
| --- |
| **Action:** Training Program dashboard |
| **Brief Description:** The user can see the exercises available, the progress on each of this exercises and performance. |
| **Actors:** Registered user |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. Initially only Level 1 lessons are active. 2. The user selects the lesson that he wants to take and completes it. He is redirected to the dashboard. 3. The lessons which have been completed are indicated with green check mark on the lesson icon. 4. A lesson can be taken multiple times for practice. 5. Once all the lessons in level 1 are completed, then the level 2 lessons are unlocked. |
| **Extensions:** |
| **Post Conditions:** |
| **Special Requirements:** The user needs to maintain a good score to progress further. If the performance on the lessons are poor, then the progress is reset, and the user needs to take the lessons from the beginning. |

|  |
| --- |
| **Action:** Lessons page |
| **Brief Description:** The user selects and take a lesson on this page. |
| **Actors:** Registered User |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. The user selects a lesson. 2. The user lands on the lesson page. 3. The lesson page will have interactive challenges. The challenges can be of textual type, multiple choice type, audio type and voice type. 4. The user will be able to track the progress he’s making on the challenges in an active lesson. 5. With each wrong answer a little “health” will be reduced from the “health bar”. 6. If the health bar hits zero, then the user would have to start the lesson from the beginning. 7. The user can choose to close a lesson in the middle of the challenge. The progress will be saved. |
| **Extensions:** |
| **Post Conditions:** The progress and the performance of each lesson is saved and indicated on the training program dashboard. |
| **Special Requirements:** |

|  |
| --- |
| **Action:** Trophy collection page |
| **Brief Description:** The user can win trophies by completing challenges. The user can see all the earned trophies on this page. |
| **Actors:** Registered User |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. The user wants to check the trophies he won. 2. Each trophy will have levels I, II and III. 3. These achievements can be shared on the social media provided the user has chosen to connect Facebook account to the application. |
| **Extensions:** |
| **Post Conditions:** |
| **Special Requirements:** |

|  |
| --- |
| **Action:** League standing page |
| **Brief Description:** Users can compare their performance with other users learning the same language. |
| **Actors:** Registered user |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. The user win experience point (XP) based on the lessons they are taking and their performance in those lessons. These XP add up to help users climb through the league standing. The user can tap on the league standing button to check his league standing. 2. Here he can see how many people are performing better than him and try to take more lessons and challenges to add more XP points. The higher a user finishes, the more chances of him to earn more trophies. 3. The user can share their league standing on Facebook. |
| **Extensions:** |
| **Post Conditions:** |
| **Special Requirements:** |

|  |
| --- |
| **Action:** Practice page |
| **Brief Description:** The user can practice the lessons that he has taken so far. |
| **Actors:** Registered user |
| **Pre-conditions:** The user has network connectivity and a device to access the application. |
| **Basic Flow of events:**   1. The user can choose to practice the lessons they have taken so far. 2. On this page there will be miscellaneous challenges from all the completed lessons. 3. The user can earn XP by taking more practice challenges. 4. The user can lose “health” from their “Health bar” with wrong answers. |
| **Extensions:** |
| **Post Conditions:** |
| **Special Requirements:** |

7. Work Breakdown Structure

7.1 Overall WBS

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **WBS Code** | **Element Name** | **Description** |
| 1 | 1 | Digital Language Training System | All work to implement a multi-platform digital language training system that engages the user through a cognitively aware gamified interface. |
| 2 | 1.1 | Initiation | The work to initiate project. |
| 3 | 1.1.1 | Develop Business Case | Project sponsor creates and owns the Business Case. |
| 3 | 1.1.2 | Assign a Project Manager | Assign a Project manager who will develop the Project Overview Statement. |
| 3 | 1.1.3 | Evaluation and Recommendations | The project manager will evaluate the business case and give valid recommendations. |
| 3 | 1.1.4 | Develop Project Overview Statement | Project Manager develops a project overview statement, or a project charter based on the approved business case document. |
| 3 | 1.1.5 | Project overview statement review | The project sponsor reviews the POS document and makes recommendations. |
| 3 | 1.1.6 | Sign off Project Overview Statement | The project manager works on the POS document based on the recommendations from the project sponsor and gets the sign off on the final document. |
| 2 | 1.2 | Planning | The work for the planning phase of the project. |
| 3 | 1.2.1 | Create Preliminary Scope Statement | Project manager creates a preliminary scope document. |
| 3 | 1.2.2 | Determine Core Project Team | The project manager determines the Core Project Team and requests resources. This team includes functional managers, technical managers, business analysts, expert tutors, language experts, designers and team leaders. |
| 3 | 1.2.3 | Project kick-off meeting with the core team | The planning is officially starting. The project manager and the Core Project Team meet and discusses on the business vision, the project vision, the strategy, roles and scope of responsibility, team building and team commitments and ground rules. |
| 3 | 1.2.4 | Develop Project Plan | Under the direction of the Project Manager the Core Project Team develops the project plan. |
| 4 | 1.2.4.1 | Develop Work Breakdown Structure | The Core Project Team develops the WBS. |
| 4 | 1.2.4.2 | Develop Project Schedule | The Core Project Team analyzes all the work under their ownership and estimates the amount of time each work needs to be completed and the milestones. |
| 4 | 1.2.4.3 | Develop budget plan | The Core Project Team analyses all the work under their ownership and estimates the amount of resources needed to complete them and comes up with a budget to create the budget plan. |
| 4 | 1.2.4.4 | Develop management plan | Once the scope, schedule, and cost baselines have been established, Core Project Team create the steps in change control process the team will take to manage variances to these plans. All these management plans usually include a review and approval process for modifying the baselines. |
| 4 | 1.2.4.5 | Developing resources plan | The Core Project Team analyses the resources in hand and also if any more resources that needs to add and creates a staffing plan. The staffing plan is a chart that shows the time periods, usually month, quarter, year, that each resource will come onto and leave the project. |
| 4 | 1.2.4.6 | Develop risk management plan | The Core Project Team develops risk management plan to understand and communicate how the team will respond to the high-risk events. |
| 4 | 1.2.4.7 | Develop quality assurance plan | The Software Quality Assurance Team prepares the standards, practices and metrics are used during the overall SQA process. |
| 4 | 1.2.4.8 | Develop Acceptance plan | The Core Project Team develops the schedule of tasks and deliverables, standards and metrics that are required to gain the customers’ acceptance. |
| 4 | 1.2.4.9 | Develop Communications Plan | The Core Project Team develops the communication plan to identify the communication goals, stakeholders and strategies, activities and timeframes and to improve the communication process. |
| 4 | 1.2.4.10 | Development procurement plan | Core Project Team develops the procurement plan identifying the products and services obtained from external suppliers and the tasks involved in procuring the products. |
| 3 | 1.2.5 | Complete project plan covering all the aspects of the project. | Project Manager submits the project plan for approval. |
| 3 | 1.2.6 | Project plan approval | The project plan is approved, and the Project Manager Approval has permission to proceed to execute the project according to the project plan. |
|  |  |  |  |
| 2 | 1.3 | Launch | All work associated with application launch is provided within section 1.3. |
| 3 | 1.3.1 | Application Launch Kick-Off Meeting | Project manager meets with client, Business Team and Development Team. |  |  |  |
| 4 | 1.3.1.1 | Gather USER Requirements | Requirements for font end application Views, application workflows, back end data management and business requirements are collated and presented by Project Manager and Core Project Team to stakeholders. |  |  |  |
| 4 | 1.3.1.2 | Requirement Review | Stakeholder analysts and Core Project Team developers make any necessary adjustments to requirements. |  |  |  |
| 4 | 1.3.1.3 | Technical Analysis | Project Manager, stakeholder analysts and Development Team review functionalities of each requirement in terms of their execution. |  |  |  |
| 4 | 1.3.1.4 | Risk Analysis | Project Manager, stakeholder analysts and Development Team review technical/operational and business risks associated with requirements. |  |  |  |
| 4 | 1.3.1.5 | Operational Analysis | Project Manager, stakeholder analysts and Development Team clarify any uncertainties related to milestones, deliverables, deadlines, etc. |
| 4 | 1.3.1.6 | Documentation of Requirements | Business Analyst Team documents all requirements finalized and agreed upon by stakeholders. |
| 4 | 1.3.1.7 | Software requirements review | Senior Business Analyst reviews documented requirements. |
| 4 | 1.3.1.8 | Software requirements document submission | Core Project Team submits Software Requirements for approval. |
| 4 | 1.3.1.9 | Software requirements approval | Core Project Team is cleared to begin with the design of the project according to Software Requirements document. |
|  |  |  |  |
| 3 | 1.3.2 | Application launch |  |
| 4 | 1.3.2.1 | Splash Screen Design | UI Designers sketch splash screen and present to Business Analyst and System Architect, Security Team and Development Team |
| 5 | 1.3.2.1.1 | Splash Screen Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.1.2 | Splash Screen Design Sign Off | Project Manager and Client Representative sign off on final Splash Screen Design for Development. |
| 5 | 1.3.2.1.3 | Splash Screen Programmed into Front End. | Developers Create Splash Screen View for all Devices according to Design |
| 5 | 1.3.2.1.4 | Splash Screen Tested | Security Team and Software Quality Assurance Team performs unit tests on representative sample of hardware devices. |
| 5 | 1.3.2.1.5 | Bug Identification | Bug reports and security issues found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.1.6 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.1.7 | Splash Screen Signoff | Project Manager Signs off on Splash Screen. |
| 4 | 1.3.2.2 | New User Account Setup | UI Designers sketch New User Account Setup and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.2.1 | New User Account Setup Screen Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.2.2 | New User Account Setup Screen Design Sign Off | Project Manager and Client Representative sign off on final Splash Screen Design for Development. |
| 5 | 1.3.2.2.3 | New User Account Setup Screen Programmed into Front End. | Developers Create New User Account Setup Screen View for all Devices |
| 5 | 1.3.2.2.4 | New User Account Setup Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client account data setup functions. |
| 5 | 1.3.2.2.5 | New User Account Setup functionalities programmed into Middle Layer | All methods for Instantiating or accessing User Account data via Front End Application are created. |
| 5 | 1.3.2.2.6 | New User Account Setup Unit Testing | Security Team and Software Quality Assurance Team unit test for |
| 5 | 1.3.2.2.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.2.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.2.9 | New User Account Setup Signoff | Project Manager Signs off on Splash Screen. |
| 4 | 1.3.2.3 | Training Language Selection Interface | UI Designers sketch Training Language Selection Interface and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.3.1 | Training Language Selection Interface Design | UI Designers sketch Training Language Selection Interface and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.3.2 | Training Language Selection Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.3.3 | Training Language Selection Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 5 | 1.3.2.3.4 | Training Language Selection Interface Programmed into Front End. | Developers Create Training Language Selection Screen View for all Devices |
| 5 | 1.3.2.3.5 | Training Language Selection Interface Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve Training Language Selection functions. |
| 5 | 1.3.2.3.6 | Training Language Selection functionalities programmed into Middle Layer | All methods for instantiating or accessing local/remoted data services via Front End Application for the purposes of Language Training Selection are created. |
| 5 | 1.3.2.3.7 | Training Language Selection Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.3.8 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.3.9 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.3.10 | Training Language Selection Interface Signoff | Project Manager Signs off on Training Language Selection Interface. |
| 4 | 1.3.2.4 | Language Training Program Curriculum | All long, medium and short term language learning goals are set by each Language Training Expert and those goals are used to design all curriculum based aspects of the Language Training Program. |
| 5 | 1.3.2.4.1 | Setting of Language Training Program long term goals. | Language Training Experts set the long term major milestones for each Level of Difficulty. |
| 5 | 1.3.2.4.2 | Setting of Language Training medium term goals. | Language Training Experts organize major milestones for each Level of Difficulty by topic into Learning Modules. |
| 5 | 1.3.2.4.3 | Setting of Language Training short term goals. | Language Training Experts organize curriculum into Lessons that incrementally cover all aspects of the Module Topic. |
| 5 | 1.3.2.4.4 | Design of Lesson Exercises | Language Training Experts construct Exercises that require mastery of curriculum for appropriate Lesson of appropriate Module. |
| 5 | 1.3.2.4.5 | Language Training Program Progression | Language Training Experts set expectations that will be used to set rules for satisfactory progression through each Lesson/Module/Level. |
| 5 | 1.3.2.4.6 | Language Training Program Presentation | Language Experts present theoretical rational for all long, medium and short term language learning goals to Client, Business Analyst and Core Project Teams. |
| 5 | 1.3.2.4.7 | Language Training Program Review | Client, Business Analyst, Core Project and Language Experts negotiate final expectations and constraints on Language Training Program Philosophy. |
| 5 | 1.3.2.4.8 | Language Training Program Signoff | Project Manager and Client signoff on Language Training Program. |
| 4 | 1.3.2.5 | Language Training Program Interface |  |
| 5 | 1.3.2.5.1 | Language Training Program Interface Design | UI Designers sketch Language Training Program Interface and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.5.2 | Language Training Program Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.5.3 | Language Training Program Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 5 | 1.3.2.5.4 | Language Training Program Interface Programmed into Front End. | Developers Create Language Training Progam View for all Devices |
| 5 | 1.3.2.5.5 | Language Training Program Interface Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve Language Training Program functions related to viewing progress within the training program via Modules marked as complete and highlighted edges along the recommended learning path. |
| 5 | 1.3.2.5.6 | Language Training Program functionalities programmed into Middle Layer | All methods for instantiating or accessing local/remoted data services via Front End Application for the purposes of Language Training Program navigation and Module selection are created. |
| 5 | 1.3.2.5.7 | Language Training Program Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.5.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.5.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.5.9 | Language Training Program Interface Signoff | Project Manager and Client signoff on Language Training Program interface. |
| 4 | 1.3.2.6 | Language Training Modules |  |
| 5 | 1.3.2.6.1 | Language Training Module Interface Design | UI Designers sketch Language Training Program Module related Interface capabilities and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.6.2 | Language Training Module Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.6.3 | Language Training Module Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 5 | 1.3.2.6.4 | Language Training Module Interface Programmed into Front End. | Developers Create Language Training Module viewing and selection capabilities for all Devices |
| 5 | 1.3.2.6.5 | Language Training Module Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve all necessary Language Training Module related data. |
| 5 | 1.3.2.6.6 | Language Training Module functionalities programmed into Middle Layer | All methods for instantiating or accessing local/remoted data services via Front End Application for the purposes of Language Training Module selection are created. |
| 5 | 1.3.2.6.7 | Language Training Module Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.6.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.6.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.6.9 | Language Training Program Module Signoff | Project Manager and Client signoff on Language Training Program interface. |
| 4 | 1.3.2.7 | Language Training Lesson Notes Interface |  |
| 5 | 1.3.2.7.1 | Language Training Lesson Notes Interface Design | UI Designers sketch Language Training Lesson Curriculum Notes Interface and related Interface capabilities and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.7.2 | Language Training Lesson Notes Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.7.3 | Language Training Lesson Notes Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 5 | 1.3.2.7.4 | Language Training Lesson Notes Curriculum Interface Programmed into Front End. | Developers Create Language Training Lesson Curriculum Notes selection and viewing interface capabilities for all Devices |
| 5 | 1.3.2.7.5 | Language Training Lesson Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve all necessary Language Training Lesson Notes related data. |
| 5 | 1.3.2.7.6 | Language Training Lesson Notes functionalities programmed into Middle Layer | All methods for instantiating or accessing local/remoted data services via Front End Application for the purposes of Language Training Lesson Notes are created. |
| 5 | 1.3.2.7.7 | Language Training Lesson Notes Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.7.8 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.7.9 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.7.10 | Language Training Lesson Notes Front End Interface Signoff | Project Manager and Client signoff on Language Training Lesson Notes Interface |
| 4 | 1.3.2.8 | Language Training Lesson Attempt Interface |  |
| 5 | 1.3.2.8.1 | Language Training Lesson Attempt Global Components Interface Design | UI Designers sketch Language Training Lesson Attempt Global Components (Exit Lesson Button, Life Meter, Lesson Progress Bar, Exercise Body and Response Submit buttons) and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 5 | 1.3.2.8.1.1 | Language Training Lesson Attempt Global Components Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 5 | 1.3.2.8.1.2 | Language Training Lesson Attempt Global Components Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 5 | 1.3.2.8.2 | Language Training Lesson Interactive Exercises |  |
| 6 | 1.3.2.8.2.1 | Language Training Lesson Interactive Exercise: Textual Challenge/ Textual Response Front End Design | UI Designers sketch Language Training Lesson Interactive Exercise: Textual Challenge/ Textual Response and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 7 | 1.3.2.8.2.1.1 | Language Training Lesson Interactive Exercise: Textual Challenge/ Textual Response Front End Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 7 | 1.3.2.8.2.1.2 | Language Training Lesson Interactive Exercise: Textual Challenge/ Textual Response Interface Front End Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 7 | 1.3.2.8.2.1.3 | Language Training Lesson Interactive Exercise: Textual Challenge/ Textual Response Interface Front End Creation |  |
| 7 | 1.3.2.8.2.1.4 | Language Training Lesson Interactive Exercises Textual Challenge/ Textual Response Exercises populated into Remote Data Layer | Remote Data Layer creates database of Textual Challenge/ Textual Response Exercises and Answers along with back end accessor services. |
| 7 | 1.3.2.8.2.1.5 | Language Training Lesson Interactive Exercises Textual Challenge/ Textual Response Middle Layer | Middle Layer Development Team creates all methods connecting remote and local TC/TR data to front end Exercise application view. |
| 7 | 1.3.2.8.2.1.6 | Language Training Lesson Interactive Exercises Textual Challenge/ Textual Response Front End Interface Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 7 | 1.3.2.8.2.1.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 7 | 1.3.2.8.2.1.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 7 | 1.3.2.8.2.1.9 | Language Training Lesson Interactive Exercises Textual Challenge/ Textual Response Front End Interface Signoff | Project Manager and Client signoff on Language Training Lesson Interactive Exercises Textual Challenge/ Textual Response Front End Interface |
| 6 | 1.3.2.8.2.2 | Language Training Lesson Interactive Exercise Front End Interface: Textual Challenge/ Multiple Choice Response Design | UI Designers sketch Language Training Lesson Interactive Exercise: Textual Challenge/ Multiple Choice Response and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 7 | 1.3.2.8.2.2.1 | Language Training Lesson Interactive Exercise Interface: Textual Challenge/ Multiple Choice Response Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 7 | 1.3.2.8.2.2.2 | Language Training Lesson Interactive Exercise Interface: Textual Challenge/ Multiple Choice Response Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 7 | 1.3.2.8.2.2.3 | Language Training Lesson Interactive Exercise: Textual Challenge/ Multiple Choice Response Interface Front End Creation |  |
| 7 | 1.3.2.8.2.2.4 | Language Training Lesson Interactive Exercises Textual Challenge/ Multiple Choice Response Exercises populated into Remote Data Layer | Remote Data Layer creates database of Textual Challenge/ Multiple Choice Response Exercises and Answers along with back end accessor services. |
| 7 | 1.3.2.8.2.2.5 | Language Training Lesson Interactive Exercises Textual Challenge/ Multiple Choice Response Middle Layer | Middle Layer Development Team creates all methods connecting remote and local Textual Challenge/ Multiple Choice Response data to front end Exercise application view. |
| 7 | 1.3.2.8.2.2.6 | Language Training Lesson Interactive Exercises Textual Challenge/ Multiple Choice Response Front End Interface Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 7 | 1.3.2.8.2.2.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 7 | 1.3.2.8.2.2.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 7 | 1.3.2.8.2.2.9 | Language Training Lesson Interactive Exercises Textual Challenge/ Multiple Choice Response Front End Interface Signoff | Project Manager and Client signoff on Language Training Lesson Interactive Exercises Textual Challenge/ Multiple Choice Response Front End Interface |
| 6 | 1.3.2.8.2.3 | Language Training Lesson Interactive Exercise Interface: Audio Challenge/ Textual Response Design | UI Designers sketch Language Training Lesson Interactive Exercise: Audio Challenge/ Textual Response and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 7 | 1.3.2.8.2.3.1 | Language Training Lesson Interactive Exercise Interface: Audio Challenge/ Textual Response Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 7 | 1.3.2.8.2.3.2 | Language Training Lesson Interactive Exercise Interface: Audio Challenge/ Textual Response Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 7 | 1.3.2.8.2.3.3 | Language Training Lesson Interactive Exercise: Audio Challenge/ Textual Response Interface Front End Creation |  |
| 7 | 1.3.2.8.2.3.4 | Language Training Lesson Interactive Exercises Audio Challenge/ Textual Response Exercises populated into Remote Data Layer | Remote Data Layer creates database of Audio Challenge/ Textual Response Exercises and Answers along with back end accessor services. |
| 7 | 1.3.2.8.2.3.5 | Language Training Lesson Interactive Exercises Audio Challenge/ Textual Response Middle Layer | Middle Layer Development Team creates all methods connecting remote and local Audio Challenge/ Textual Response data to front end Exercise application view. |
| 7 | 1.3.2.8.2.3.6 | Language Training Lesson Interactive Exercises Audio Challenge/ Textual Response Front End Interface Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 7 | 1.3.2.8.2.3.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 7 | 1.3.2.8.2.3.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 7 | 1.3.2.8.2.3.9 | Language Training Lesson Interactive Exercises Audio Challenge/ Textual Response Front End Interface Signoff | Project Manager and Client signoff on Language Training Lesson Interactive Exercises Audio Challenge/ Textual Response Front End Interface |
| 6 | 1.3.2.8.2.4 | Language Training Lesson Interactive Exercise Interface: Textual Challenge/ Audio Response Design | UI Designers sketch Language Training Lesson Interactive Exercise: Textual Challenge/ Audio Response and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 7 | 1.3.2.8.2.4.1 | Language Training Lesson Interactive Exercise Interface: Textual Challenge/ Audio Response Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 7 | 1.3.2.8.2.4.2 | Language Training Lesson Interactive Exercise Interface: Textual Challenge/ Audio Response Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 7 | 1.3.2.8.2.4.3 | Language Training Lesson Interactive Exercise: Textual Challenge/ Audio Response Interface Front End Creation |  |
| 7 | 1.3.2.8.2.4.4 | Language Training Lesson Interactive Exercises Textual Challenge/ Audio Response Exercises populated into Remote Data Layer | Remote Data Layer creates database of Textual Challenge/ Audio Response Exercises and Answers along with back end accessor services. |
| 7 | 1.3.2.8.2.4.5 | Language Training Lesson Interactive Exercises Textual Challenge/ Audio Response Middle Layer | Middle Layer Development Team creates all methods connecting remote and local Textual Challenge/ Audio Response data to front end Exercise application view. |
| 7 | 1.3.2.8.2.4.6 | Language Training Lesson Interactive Exercises Textual Challenge/ Audio Response Front End Interface Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 7 | 1.3.2.8.2.4.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 7 | 1.3.2.8.2.4.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 7 | 1.3.2.8.2.4.9 | Language Training Lesson Interactive Exercises Textual Challenge/ Audio Response Front End Interface Signoff | Project Manager and Client signoff on Language Training Lesson Interactive Exercises Textual Challenge/ Audio Response Front End Interface |
| 6 | 1.3.2.8.2.5 | Language Training Lesson Interactive Exercise Interface: Audio Challenge/ Audio Response Design | UI Designers sketch Language Training Lesson Interactive Exercise: Audio Challenge/ Audio Response and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 7 | 1.3.2.8.2.5.1 | Language Training Lesson Interactive Exercise Interface: Audio Challenge/ Audio Response Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 7 | 1.3.2.8.2.5.2 | Language Training Lesson Interactive Exercise Interface: Audio Challenge/ Audio Response Interface Design Sign Off | Project Manager and Client Representative sign off on final design for Development. |
| 7 | 1.3.2.8.2.5.3 | Language Training Lesson Interactive Exercise: Audio Challenge/ Audio Response Interface Front End Creation |  |
| 7 | 1.3.2.8.2.5.4 | Language Training Lesson Interactive Exercises Audio Challenge/ Audio Response Exercises populated into Remote Data Layer | Remote Data Layer creates database of Audio Challenge/ Audio Response Exercises and Answers along with back end accessor services. |
| 7 | 1.3.2.8.2.5.5 | Language Training Lesson Interactive Exercises Audio Challenge/ Audio Response Middle Layer | Middle Layer Development Team creates all methods connecting remote and local Audio Challenge/ Audio Response data to front end Exercise application view. |
| 7 | 1.3.2.8.2.5.6 | Language Training Lesson Interactive Exercises Audio Challenge/ Audio Response Front End Interface Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 7 | 1.3.2.8.2.5.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 7 | 1.3.2.8.2.5.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 7 | 1.3.2.8.2.5.9 | Language Training Lesson Interactive Exercises Audio Challenge/ Audio Response Front End Interface Signoff | Project Manager and Client signoff on Language Training Lesson Interactive Exercises Audio Challenge/ Audio Response Front End Interface |
| 4 | 1.3.2.9 | Health Status |  |
| 5 | 1.3.2.9.1 | Gamification Design of Health Status | Gamification Designers Design Health Status attributes and qulaifications for gaining/losing Health Points |
| 6 | 1.3.2.9.1.1 | Gamification Design Review for Health Status | Business Analyst, System Architect, and Development Team negotiate final Gamification of Health Status Design with UI Designers according to business and software constraints. |
| 6 | 1.3.2.9.1.2 | Gamification Design Weekly League Competitions Signoff | Project Manager and Client Representative sign off on final Gamification Health Status Design for Development. |
| 5 | 1.3.2.9.2 | Health Status Interface Component Setup | UI Designers sketch Health Status Front End Components and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.9.2.1 | Health Status Interface Component Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 6 | 1.3.2.9.2.2 | Health Status Interface Component Design Sign Off | Project Manager and Client Representative sign off on final Weekly League Standing Interface Design for Development. |
| 6 | 1.3.2.9.2.3 | Health Status Interface Components Programmed into Front End. | Developers Create Health Status Interface Components for all appropriate Views for all Devices |
| 6 | 1.3.2.9.2.4 | Health Status Interface Components Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client Health Status functionalities according to gamification design of leagues. |
| 6 | 1.3.2.9.2.5 | Health Status Interface functionalities programmed into Middle Layer | All methods for Instantiating or accessing Heath Status related data via Front End Application are created. |
| 6 | 1.3.2.9.2.6 | Health Status Interface Components Setup Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 6 | 1.3.2.9.2.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 6 | 1.3.2.9.2.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 6 | 1.3.2.9.2.9 | Weekly League Standing Interface Account Setup Signoff | Project Manager Signs off on Health Status Interface components. |
| 4 | 1.3.2.10 | Weekly XP Points |  |
| 5 | 1.3.2.10.1 | Gamification Design of Weekly XP Points | Gamification Designers Design Weekly XP Points attributes and qulaifications for earning Weekly XP Points |
| 6 | 1.3.2.10.1.1 | Gamification Design Review for Weekly XP Points | Business Analyst, System Architect, and Development Team negotiate final Gamification of Weekly XP Points Design with UI Designers according to business and software constraints. |
| 6 | 1.3.2.10.1.2 | Gamification Design Weekly XP Points Signoff | Project Manager and Client Representative sign off on final Gamification Weekly XP Points Design for Development. |
| 5 | 1.3.2.10.2 | Weekly XP Points Interface Component Setup | UI Designers sketch Weekly XP Points Front End Components and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.10.2.1 | Weekly XP Points Interface Component Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 6 | 1.3.2.10.2.2 | Weekly XP Points Interface Component Design Sign Off | Project Manager and Client Representative sign off on final Weekly XP Points Interface Componentss Design for Development. |
| 6 | 1.3.2.10.2.3 | Weekly XP Points Interface Components Programmed into Front End. | Developers create Weekly XP Points Interface Components for all appropriate Views for all Devices |
| 6 | 1.3.2.10.2.4 | Weekly XP Points Interface Components Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client Weekly XP Points functionalities according to gamification design. |
| 6 | 1.3.2.10.2.5 | Weekly XP Points Interface Components functionalities programmed into Middle Layer | All methods for Instantiating or accessing Weekly XP Points related data via Front End Application are created. |
| 6 | 1.3.2.10.2.6 | Weekly XP Points Interface Components Setup Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 6 | 1.3.2.10.2.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 6 | 1.3.2.10.2.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 6 | 1.3.2.10.2.9 | Weekly XP Points Interface Signoff | Project Manager Signs off on Weekly XP Points Interface components. |
| 4 | 1.3.2.11 | League Standing |  |
| 5 | 1.3.2.11.1 | Gamification Design of Weekly Leagues | Gamification Designers Design Weekly League attributes and qulaifications for Promotion/Demotion to subsequent Leagues |
| 6 | 1.3.2.11.1.1 | Gamification Design Review for Weekly League Competitions | Business Analyst, System Architect, and Development Team negotiate final Gamification of Weekly League Competition Design with UI Designers according to business and software constraints. |
| 6 | 1.3.2.11.1.2 | Gamification Design Weekly League Competitions Signoff | Project Manager and Client Representative sign off on final Gamification Trophy Collection Design for Development. |
| 5 | 1.3.2.11.2 | Weekly League Standing Interface Setup | UI Designers sketch Weekly League Standing Front End and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.11.2.1 | Weekly League Standing Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 6 | 1.3.2.11.2.2 | Weekly League Standing Interface Design Sign Off | Project Manager and Client Representative sign off on final Weekly League Standing Interface Design for Development. |
| 6 | 1.3.2.11.2.3 | Weekly League Standing Interface Programmed into Front End. | Developers Create Weekly League Standing Interface View for all Devices |
| 6 | 1.3.2.11.2.4 | Weekly League Standing Interface Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client Weekly League Standing functions according to gamification design of leagues. |
| 6 | 1.3.2.11.2.5 | Weekly League Standing Interface functionalities programmed into Middle Layer | All methods for Instantiating or accessing Weekly League Standing Interface data via Front End Application are created. |
| 6 | 1.3.2.11.2.6 | Weekly League Standing Interface Setup Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 6 | 1.3.2.11.2.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 6 | 1.3.2.11.2.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 6 | 1.3.2.11.2.9 | Weekly League Standing Interface Account Setup Signoff | Project Manager Signs off on Weekly League Standing Interface. |
| 4 | 1.3.2.12 | Trophy Collection |  |
| 5 | 1.3.2.12.1 | Gamification Design of Trophy Collection | Language Experts and Gamification Designers Design Trophy entities and qulaifications for Trophy attainment |
| 6 | 1.3.2.12.1.1 | Gamification Design Review for Trophy Collection | Business Analyst, System Architect, and Development Team negotiate final Gamification of Trophy Collection Design with UI Designers according to business and software constraints. |
| 6 | 1.3.2.12.1.2 | Gamification Design Trophy Collection Signoff | Project Manager and Client Representative sign off on final Gamification Trophy Collection Design for Development. |
| 5 | 1.3.2.12.2 | Trophy Collection Interface Setup | UI Designers sketch Trophy Collection Front End and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.12.2.1 | Trophy Collection Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 6 | 1.3.2.12.2.2 | Trophy Collection Interface Design Sign Off | Project Manager and Client Representative sign off on final Trophy Collection Interface Design for Development. |
| 5 | 1.3.2.12.3 | Trophy Collection Interface Programmed into Front End. | Developers Create Trophy Collection Interface View for all Devices |
| 5 | 1.3.2.12.4 | Trophy Collection Interface Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client Gamification and Trophy Collection Interface functions. |
| 5 | 1.3.2.12.5 | Trophy Collection Interface functionalities programmed into Middle Layer | All methods for Instantiating or accessing Trophy Collection Interface data via Front End Application are created. |
| 5 | 1.3.2.12.6 | Trophy Collection Interface Setup Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.12.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.12.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.12.9 | Trophy Collection Interface Setup Signoff | Project Manager Signs off on Trophy Collection Interface. |
| 4 | 1.3.2.13 | Rewards Points |  |
| 5 | 1.3.2.13.1 | Gamification Design of Rewards Points | Gamification Designers Design Rewards Points and qulaifications for earning them |
| 6 | 1.3.2.13.1.1 | Gamification Design Review for Rewards Points | Business Analyst, System Architect, and Development Team negotiate final Gamification of Trophy Collection Design with UI Designers according to business and software constraints. |
| 6 | 1.3.2.13.1.2 | Gamification Design Rewards Points Signoff | Project Manager and Client Representative sign off on final Gamification Rewards Points Design for Development. |
| 5 | 1.3.2.13.2 | Rewards Points Interface Components Setup | UI Designers sketch Rewards Points Front End Components and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.13.2.1 | Rewards Points Interface Components Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 6 | 1.3.2.13.2.2 | Rewards Points Interface Components Design Sign Off | Project Manager and Client Representative sign off on final Rewards Points Interface Components Design for Development. |
| 5 | 1.3.2.13.3 | Rewards Points Interface Components Programmed into Front End. | Developers Create Rewards Points Interface Components Views for all Devices |
| 5 | 1.3.2.13.4 | Rewards Points Interface Components Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client Gamification and Rewards Points Interface Components functions. |
| 5 | 1.3.2.13.5 | Rewards Points Interface functionalities programmed into Middle Layer | All methods for Instantiating or accessing Rewards Points Interface Component data via Front End Application are created. |
| 5 | 1.3.2.13.6 | Rewards Points Interface Components Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.13.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.13.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.13.9 | Rewards Points Interface Components Setup Signoff | Project Manager Signs off on Rewards Points Interface Components. |
| 4 | 1.3.2.14 | App Store |  |
| 5 | 1.3.2.14.1 | Gamification Design of App Store for spending Rewards Points | Gamification Designers Design App Store Privileges |
| 6 | 1.3.2.14.1.1 | Gamification Design Review for App Store | Business Analyst, System Architect, and Development Team negotiate final Gamification of App Store Privileges Design with UI Designers according to business and software constraints. |
| 6 | 1.3.2.14.1.2 | Gamification Design App Store Signoff | Project Manager and Client Representative sign off on final Gamification App Store Design for Development. |
| 5 | 1.3.2.14.2 | App Store Front end Interface Setup | UI Designers sketch Rewards Points Front End App Store view(s) and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.14.2.1 | App Store Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 6 | 1.3.2.14.2.2 | App Store Interface Components Design Sign Off | Project Manager and Client Representative sign off on final App Store Interface Design for Development. |
| 5 | 1.3.2.14.3 | App Store Interface Programmed into Front End. | Developers Create App Store Interface Views for all Devices |
| 5 | 1.3.2.14.4 | App Store Interface Components Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve client Gamification and Rewards Points and App Store Interface functions. |
| 5 | 1.3.2.14.5 | App Store Interface functionalities programmed into Middle Layer | All methods for Instantiating or accessing Rewards Points for purchasing Privileges within the App Store Interface via Front End Application are created. |
| 5 | 1.3.2.14.6 | App Store Interface Unit Testing | Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.14.7 | Bug and Security Issue Identification | Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.14.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.14.9 | App Store Interface Setup Signoff | Project Manager Signs off on App Store Interface. |
| 4 | 1.3.2.15 | Cognizant Tutor |  |
| 5 | 1.3.2.15.1 | Cognizant Tutor Product Design | Language Experts present theoretical requirements of Cognizant Tutor to Project Manager, Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer, Machine Learning and Front End Development Team. |
| 6 | 1.3.2.15.1.1 | Cognizant Tutor Product Design Review | Business Analyst, System Architect, and Development Team negotiate final Cognizant Tutor Product Design with Language Experts to business and software constraints. |
| 6 | 1.3.2.15.1.2 | Cognizant Tutor Product Design Signoff | Project Manager and Client Representative sign off on final Cognizant Tutor Product Design for Development. |
| 5 | 1.3.2.15.2 | Cognizant Tutor Front end Interface Setup | UI Designers sketch Cognizatn Tutor Front End components and present to Business Analyst, System Architect, Security Team, Local/Remote Data Layer Team, Middle Layer and Front End Development Team. |
| 6 | 1.3.2.15.2.1 | Cognizant Tutor Interface Design Review | Business Analyst, System Architect, Security Team and Development Team negotiate final UI Design with UI Designers according to business, software, and security constraints. |
| 36 | 1.3.2.15.2.2 | Cognizant Tutor Interface Components Design Sign Off | Project Manager and Client Representative sign off on final Cognizant Tutor Interface Design for Development. |
| 35 | 1.3.2.15.3 | Cognizant Tutor Interface Programmed into Front End. | Developers Create App Store Interface Views for all Devices |
| 5 | 1.3.2.15.4 | App Store Interface Components Local/Remote Data | Local and Remote Data Layer Teams create databases and database services that serve Cognizant Tutor functions. |
| 5 | 1.3.2.15.5 | Cognizant Tutor functionalities programmed into Back End | All methods for Cognizant Tutor training are created. |
| 5 | 1.3.2.15.6 | App Store Interface Unit Testing | Language Experts, Security Team and Software Quality Assurance Team unit test for bugs and security issues |
| 5 | 1.3.2.15.7 | Interaction Deficiencies, Bug and Security Issue Identification | Interaction Deficiencies, Bug and security issue reports found during Unit Testing are sent to appropriate Development Team Leads |
| 5 | 1.3.2.15.8 | Fix Report and Code Update | All fixes are documented, updated and re-tested in the Unit Testing Phase. |
| 5 | 1.3.2.15.9 | Cognizant Tutor Interface Setup Signoff | Project Manager Signs off on Cognizant Tutor Interface. |
| 3 | 1.3.3 | Integration Testing | System Test team applies use cases to full functionality of Application across all modules. |
| 3 | 1.3.4 | Performance Testing | Performance Test team applies performance-based use case tests according to Business Team and Stake Holder expectations. |
| 3 | 1.3.5 | Security Testing | Security Team applies use case-based security tests to ensure Security Policies are all met or exceeded. |
| 2 | 1.4 | Beta Release | User evaluations administered and reviewed by Product Support Team. Product Support Team will assist users in any subsequent evaluations. |
| 3 | 1.4.1 | Enhance Beta | User evaluation-based enhancements deemed congruent with Business Team and Project Team are implemented. |
| 3 | 1.5 | Final Release | Application is distributed to market. |

* 1. Association with RBS

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement number** | **Module** | **Functions** | **Task Number** |
| 3.0 | Application launch and entry point | Splash Screen | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.5  1.3.6  1.3.9  1.3.9.1  3.10.2.1  1.3.11  1.3.11.1  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
|  |  | New User Account Setup |  |
| 3.1 | Language selection interface | Present all Language Training Programs and functionalities for selecting a particular program. | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.6  1.3.8  1.3.9  1.3.9.1  1.3.10.1  1.3.10.2  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| 3.2.1 | Training Program Conceptual Organization | Difficulty Levels | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
|  |  | Modules |  |
|  |  | Lessons |  |
|  |  | Lesson Notes |  |
|  |  | Exercises |  |
| 3.2.2 | Training program progression concept | Attempts | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| 3.0 | Application launch and entry point | Complettion | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.5  1.3.6  1.3.9  1.3.9.1  3.10.2.1  1.3.11  1.3.11.1  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| XP Points Earned |
| 3.2.3 | Graphical representation of training program | Training Program graphs by level of difficulty | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.10.2  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| 3.2.1  3.2.4 | Training Program Conceptual Organization  Training Program Interface Functions and Behaviors | Modules | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.4  1.3.10.2  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Edges |
| Completed Modules |
| Edges between completed Modules |
| Module Selection |
| 3.2.2 | Training program progression concept | Textual information related to Lesson | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Access Lesson Notes |
| Start new Lesson Attempt |
| 3.2.5 | Training Program Lesson | Lesson Attempt Interface | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.4  1.3.6  1.3.8  1.3.9  1.3.9.1  1.3.10.2  1.3.12.1  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Lesson Global Elements |
| Interactive Exercises |
| 3.2.4  3.3  3.4 | Training Program Interface Functions and Behaviors  Trophy Collection Interface  League Standing Interface | Exercise Tutor | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.4  1.3.10.2  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.4  1.3.9  1.3.9.1  1.3.10.2.6  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.9  1.3.9.1 1.3.10.2.4  1.3.11.2  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Problematic Question Flag |
| Trophys |
| League Standing |
| Promotion Zone |
| Demotion Zone |
| 3.5  3.6  3.7 | Health and Practice Interface  Weekly XP Tracking  Rewards Points | View Health Status | 1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.8  1.3.9  1.3.9.1  1.3.10.1  1.3.11  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.8  1.3.9  1.3.9.1  1.3.10.1  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Earn Health Through Parctice Exercises |
| XP Points earned after each completed Lesson |
| Weekly XP points tracked for weekly League Competitions |
| Pick A Card Game |
| 3.3 | Trophy Collection Interface | Option for Advertisement and additional round of Pick A Card | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.4  1.3.9  1.3.9.1  1.3.10.2.6  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| 3.4  3.8  3.9 | League Standing Interface  Advertisements  App store interface | Optional additional round of Pick A Card | 1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.9  1.3.9.1 1.3.10.2.4  1.3.11.2  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.8  1.3.9  1.3.9.1  1.3.10.1  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.1  1.3.1.2  1.3.1.3  1.3.1.4  1.3.1.5  1.3.1.6  1.3.1.7  1.3.1.8  1.3.1.9  1.3.1.10  1.3.3  1.3.4  1.3.8  1.3.9  1.3.9.1  1.3.10.1  1.3.11.5  1.3.12  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Advertisement played |
| Display Privileges available for purchase |
| 3.5  3.10 | Health and Practice Interface  Cognizant tutor | Provide means to transact Rewards Points from Privileges | 1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24  1.3.13  1.3.14  1.3.15  1.3.16  1.3.17  1.3.18  1.3.19  1.3.20  1.3.21  1.3.22  1.3.23  1.3.24 |
| Cognizant Tutor Data Tracking, Association Rule Detection and User Messaging |
|  |  |  |  |
|  |
|  |  |  |  |
|  |
|  |
|  |  |  |  |
|  |  |  |  |
|  |
|  |  |  |  |

8. Resources and Time/Effort Estimations

|  |  |
| --- | --- |
| Team | Members |
| Core Project | Project Manager  Business Lead  Solution Architect  Technical Lead  Business Analyst  Quality Assurance Lead |
| Software Quality Assurance | Quality Assurance Lead  Testers  Functional  System  Performance  Integration |
| Business | Project Manager  Business Analyst  Business Lead |
| Design | UI Design  System Architect  Program Design Specialist (Language Specific)  Module Design Specialist (Language Specific)  Lesson Design Specialist (Language Specific)  Exercise Design Specialist (Language Specific)  Gamification Design |
| Development | Application Front End Developers  Application Middle Layer Developers  AI developers |
| Local Data Layer Team | Conceptual/Physical Database Designers  Data Services  Data Utilities  Data Security |
| Remote Data Layer Team | Conceptual/Physical Database Designers  Data Services  Data Utilities  Data Security |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task ID** | **Task #** | **Task Name** | **Estimated Days** | **Resource #** | **Resources required** |
|  | 1 | Digital Language Training System | 983 |  |  |
|  | **1.1** | **Initiation** | **11** | **1** | **Project Manager** |
| DLTS1 | 1.1.1 | Develop Business Case | 2 | 1 | Project Sponsor |
| DLTS2 | 1.1.2 | Assign Project Manager | 1 | 1 | Project Sponsor |
| DLTS3 | 1.1.3 | Evaluation and Recommendation | 2 | 1 | Project Manager |
| DLTS4 | 1.1.4 | Develop Project Overview Statement | 3 | 1 | Project Manager |
| DLTS5 | 1.1.5 | Project Overview Statement review | 2 | 1 | Project Sponsor |
| DLTS6 | 1.1.6 | Project Overview Statement sign off | 1 | 1 | Project Manager and Project Sponsor |
|  | **1.2** | **Planning** | **31** |  | **Core team** |
| DLTS7 | 1.2.1 | Create Preliminary Scope Statement | 1 | 1 | Project Manager |
| DLTS8 | 1.2.2 | Determine Core Project Team | 1 | 1 | Project Manager |
| DLTS9 | 1.2.3 | Project Core Team Kickoff meeting | 1 |  | Core Team |
|  | 1.2.4 | Develop Project Plan | 25 |  | Core Team |
| DLTS10 | 1.2.4.1 | Develop Work Breakdown Structure | 3 |  | Core Team |
| DLTS11 | 1.2.4.2 | Develop Project Schedule | 2 |  | Core Team |
| DLTS12 | 1.2.4.3 | Develop Budget Plan | 3 |  | Core Team |
| DLTS13 | 1.2.4.4 | Develop Management Plan | 2 |  | Core Team |
| DLTS14 | 1.2.4.5 | Develop Resource Plan | 2 |  | Core Team |
| DLTS15 | 1.2.4.6 | Develop Risk Plan | 5 |  | Core Team |
| DLTS16 | 1.2.4.7 | Develop Quality Assurance plan | 4 |  | QA Lead and Project Manager |
| DLTS17 | 1.2.4.8 | Develop Acceptance plan | 2 |  | Core Team |
| DLTS18 | 1.2.4.9 | Develop Communication plan | 2 |  | Core Team |
| DLTS19 | 1.2.5 | Complete project plan covering all the aspects of the project | 2 |  | Core team |
| DLTS20 | 1.2.6 | Project Plan Approval | 1 |  | Project Manager |
|  | 1.3 | Execution |  |  | Core team, Software QA Team, Business Team, Design Team, Development Team, Local Data Layer Team and Remote Data Layer Team |
|  | 1.3.1 | Application Launch Kickoff Meeting | 23 |  |  |
| DLTS21 | 1.3.1.1 | Gather User Requirements | 5 |  | Core Project Team  Business Team |
| DLTS22 | 1.3.1.2 | Requirement Review | 2 |  | Core Project Team  Business Team |
| DLTS23 | 1.3.1.3 | Technical Analysis | 3 |  | Core Project Team  Development Team  Quality Assurance Team |
| DLTS24 | 1.3.1.4 | Risk Analysis | 3 |  | Core Project Team  Development Team  Quality Assurance Team |
| DLTS25 | 1.3.1.5 | Operational Analysis | 2 |  | Core Project Team  Development Team  Quality Assurance Team |
| DLTS26 | 1.3.1.6 | Documentation of Requirements | 4 |  | Business Team |
| DLTS27 | 1.3.1.7 | Software Requirements Review | 2 |  | Business Team |
| DLTS28 | 1.3.1.8 | Software Requirements Document Submission | 1 |  | Business Team |
| DLTS29 | 1.3.1.9 | Software Requirements Document | 1 |  | Core Project Team |
|  | 1.3.2 | Language Training Program Design | 141 |  |  |
| DLTS30 | 1.3.2.1 | Language Training Program Organization | 30 |  | Design Team |
| DLTS31 | 1.3.2.2 | Language Lesson Notes Creation | 45 |  | Design Team |
| DLTS32 | 1.3.2.3 | Lesson Exercise Design | 45 |  | Design Team |
| DLTS33 | 1.3.2.4.3 | User Data Back End Conceptualization | 21 |  | Remote Data Layer Team |
| DLTS34 | 1.3.3 | System Architecture Design | 14 |  | Design Team |
| DLTS35 | 1.3.4 | Front End View UI Design | 30 |  | Design Team |
| DLTS36 | 1.3.5 | Hardware Specific Constraints | 5 |  | Design Team  Development Team |
| DLTS37 | 1.3.6 | Software Specific Design Constraints | 5 |  | Design Team  Development Team |
| DLTS38 | 1.3.8 | Local/Remote Application Layer Communications | 21 |  | Design Team  Remote Data Layer Development Team  Local Data Layer Development Team |
| DLTS39 | 1.3.9 | Design Review | 3 |  | Core Project Team |
| DLTS40 | 1.3.9.1 | Design Approval | 1 |  | Core Project Team |
|  | 1.3.10 | Application Development | 175 |  |  |
| DLTS41 | 1.3.10.1 | Application Back End Development | 60 |  | Remote Data Layer Team  Local Data Layer Team  Software Quality Assurance Team |
|  | 1.3.10.2 | Application Front End Development | 115 |  |  |
| DLTS42 | 1.3.10.2.1 | New User Account Setup | 15 |  | Development Team  Software Quality Assurance Team |
| DLTS43 | 1.3.10.2.2 | Language Training Program Module/Lesson/Exercise Interface | 60 |  | Development Team  Software Quality Assurance Team |
| DLTS44 | 1.3.10.2.3 | Social Media Interface | 8 |  | Development Team  Software Quality Assurance Team |
| DLTS45 | 1.3.10.2.4 | League Competition Interface | 8 |  | Development Team  Software Quality Assurance Team |
| DLTS46 | 1.3.10.2.5 | App Store Interface | 8 |  | Development Team  Software Quality Assurance Team |
| DLTS47 | 1.3.10.2.6 | Trophy Room Interface | 8 |  | Development Team  Software Quality Assurance Team |
| DLTS48 | 1.3.10.2.7 | Health Status Interface | 8 |  | Development Team  Software Quality Assurance Team |
|  | 1.3.11 | Application Middle Layer | 28 |  | Development Team  Software Quality Assurance Team |
| DLTS49 | 1.3.11.1 | New User Account Setup | 4 |  | Development Team  Software Quality Assurance Team |
| DLTS50 | 1.3.11.2 | Language Training Program Module/Lesson/Exercise | 4 |  | Development Team  Software Quality Assurance Team |
| DLTS51 | 1.3.11.3 | Social Media | 4 |  | Development Team  Software Quality Assurance Team |
| DLTS52 | 1.3.11.4 | League Competition | 4 |  | Development Team  Software Quality Assurance Team |
| DLTS53 | 1.3.11.5 | App Store | 4 |  | Development Team  Software Quality Assurance Team |
| DLTS54 | 1.3.11.6 | Trophy Room | 4 |  | Development Team  Software Quality Assurance Team |
| DLTS55 | 1.3.11.7 | Health Status | 4 |  | Development Team  Software Quality Assurance Team |
|  | 1.3.12 | Data Layer | 135 |  |  |
| DLTS56 | 1.3.12.1 | Training Program Back End Development | 60 |  | Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |
| DLTS57 | 1.3.12.2 | Training Program Exercise Bank Development | 45 |  | Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |
| DLTS58 | 1.3.12.3 | Training Program Exercise banck back end Services conceptualization | 30 |  | Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |
| DLTS59 | 1.3.13 | Code Review and Sign Off | 20 |  | Development Team  Remote Data Layer Team |
| DLTS60 | 1.3.14 | Software Testing | 20 |  | Software Quality Assurance Team |
| DLTS61 | 1.3.15 | Unit Testing | 30 |  | Software Quality Assurance Team |
| DLTS62 | 1.3.16 | Bug Identification | 15 |  | Development Team  Remote Data Layer Team |
| DLTS63 | 1.3.17 | Bug Fixing | 60 |  | Development Team  Remote Data Layer Team  Local Data Layer Team |
| DLTS64 | 1.3.18 | Fix Report and Code Update | 30 |  | Development Team  Remote Data Layer Team  Local Data Layer Team |
| DLTS65 | 1.3.19 | Integration Testing | 14 |  | Software Quality Assurance Team |
| DLTS66 | 1.3.20 | Performance Testing | 30 |  | Software Quality Assurance Team |
| DLTS67 | 1.3.21 | Security Testing | 21 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |
| DLTS68 | 1.3.22 | Beta Release | 30 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |
| DLTS69 | 1.3.23 | Enhanced Beta | 45 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |
| DLTS70 | 1.3.24 | Final Release | 45 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |

9. Network Diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Name** | **Duration (days)** | **Resource #** | **Resources Required** | **Predecessor Task** | **Successor Task** |
| DLTS1 | Develop Business Case | 2 | 1 | Project Sponsor |  | DLTS2 |
| DLTS2 | Assign Project Manager | 1 | 1 | Project Sponsor | DLTS1 | DLTS3 |
| DLTS3 | Evaluation and Recommendation | 2 | 1 | Project Manager | DLTS2 | DLTS4 |
| DLTS4 | Develop Project Overview Statement | 3 | 1 | Project Manager | DLTS3 | DLTS5 |
| DLTS5 | Project Overview Statement review | 2 | 1 | Project Sponsor | DLTS4 | DLTS6 |
| DLTS6 | Project Overview Statement sign off | 1 | 1 | Project Manager and Project Sponsor | DLTS5 | DLTS7, DLTS8 |
| DLTS7 | Create Preliminary Scope Statement | 1 | 1 | Project Manager | DLTS5 | DLTS9 |
| DLTS8 | Determine Core Project Team | 1 | 1 | Project Manager | DLTS5 | DLTS9 |
| DLTS9 | Project Core Team Kickoff meeting | 1 |  | Core Team | DLTS7 | DLTS10, DLTS11, DLTS12, DLTS13, DLTS14, DLTS15, DLTS16, DLTS17, DLTS18 |
|  | Develop Project Plan | 25 |  | Core Team | DLTS9 | DLTS19 |
| DLTS10 | Develop Work Breakdown Structure | 3 |  | Core Team | DLTS9 | DLTS19 |
| DLTS11 | Develop Project Schedule | 2 |  | Core Team | DLTS9 | DLTS19 |
| DLTS12 | Develop Budget Plan | 3 |  | Core Team | DLTS9 | DLTS19 |
| DLTS13 | Develop Management Plan | 2 |  | Core Team | DLTS9 | DLTS19 |
| DLTS14 | Develop Resource Plan | 2 |  | Core Team | DLTS9 | DLTS19 |
| DLTS15 | Develop Risk Plan | 5 |  | Core Team | DLTS9 | DLTS19 |
| DLTS16 | Develop Quality Assurance plan | 4 |  | QA Lead and Project Manager | DLTS9 | DLTS19 |
| DLTS17 | Develop Acceptance plan | 2 |  | Core Team | DLTS9 | DLTS19 |
| DLTS18 | Develop Communication plan | 2 |  | Core Team | DLTS9 | DLTS19 |
| DLTS19 | Complete project plan covering all the aspects of the project | 2 |  | Core team | DLTS10, DLTS11, DLTS12, DLTS13, DLTS14, DLTS15, DLTS16, DLTS17, DLTS18 | DLTS20 |
| DLTS20 | Project Plan Approval | 1 |  | Project Manager | DLTS19 | DLTS21 |
|  | Application Launch Kickoff Meeting | 23 |  |  |  |  |
| DLTS21 | Gather User Requirements | 5 |  | Core Project Team  Business Team |  |  |
| DLTS22 | Requirement Review | 2 |  | Core Project Team  Business Team |  |  |
| DLTS23 | Technical Analysis | 3 |  | Core Project Team  Development Team  Quality Assurance Team |  |  |
| DLTS24 | Risk Analysis | 3 |  | Core Project Team  Development Team  Quality Assurance Team |  |  |
| DLTS25 | Operational Analysis | 2 |  | Core Project Team  Development Team  Quality Assurance Team |  |  |
| DLTS26 | Documentation of Requirements | 4 |  | Business Team |  |  |
| DLTS27 | Software Requirements Review | 2 |  | Business Team |  |  |
| DLTS28 | Software Requirements Document Submission | 1 |  | Business Team |  |  |
| DLTS29 | Software Requirements Document | 1 |  | Core Project Team |  |  |
|  | Language Training Program Design | 141 |  |  |  |  |
| DLTS30 | Language Training Program Organization | 30 |  | Design Team |  |  |
| DLTS31 | Language Lesson Notes Creation | 45 |  | Design Team |  |  |
| DLTS32 | Lesson Exercise Design | 45 |  | Design Team |  |  |
| DLTS33 | User Data Back End Conceptualization | 21 |  | Remote Data Layer Team |  |  |
| DLTS34 | System Architecture Design | 14 |  | Design Team |  |  |
| DLTS35 | Front End View UI Design | 30 |  | Design Team |  |  |
| DLTS36 | Hardware Specific Constraints | 5 |  | Design Team  Development Team |  |  |
| DLTS37 | Software Specific Design Constraints | 5 |  | Design Team  Development Team |  |  |
| DLTS38 | Local/Remote Application Layer Communications | 21 |  | Design Team  Remote Data Layer Development Team  Local Data Layer Development Team |  |  |
| DLTS39 | Design Review | 3 |  | Core Project Team |  |  |
| DLTS40 | Design Approval | 1 |  | Core Project Team |  |  |
|  | Application Development | 175 |  |  |  |  |
| DLTS41 | Application Back End Development | 60 |  | Remote Data Layer Team  Local Data Layer Team  Software Quality Assurance Team |  |  |
|  | Application Front End Development | 115 |  |  |  |  |
| DLTS42 | New User Account Setup | 15 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS43 | Language Training Program Module/Lesson/Exercise Interface | 60 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS44 | Social Media Interface | 8 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS45 | League Competition Interface | 8 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS46 | App Store Interface | 8 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS47 | Trophy Room Interface | 8 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS48 | Health Status Interface | 8 |  | Development Team  Software Quality Assurance Team |  |  |
|  | Application Middle Layer | 28 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS49 | New User Account Setup | 4 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS50 | Language Training Program Module/Lesson/Exercise | 4 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS51 | Social Media | 4 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS52 | League Competition | 4 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS53 | App Store | 4 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS54 | Trophy Room | 4 |  | Development Team  Software Quality Assurance Team |  |  |
| DLTS55 | Health Status | 4 |  | Development Team  Software Quality Assurance Team |  |  |
|  | Data Layer | 135 |  |  |  |  |
| DLTS56 | Training Program Back End Development | 60 |  | Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |
| DLTS57 | Training Program Exercise Bank Development | 45 |  | Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |
| DLTS58 | Training Program Exercise banck back end Services conceptualization | 30 |  | Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |
| DLTS59 | Code Review and Sign Off | 20 |  | Development Team  Remote Data Layer Team |  |  |
| DLTS60 | Software Testing | 20 |  | Software Quality Assurance Team |  |  |
| DLTS61 | Unit Testing | 30 |  | Software Quality Assurance Team |  |  |
| DLTS62 | Bug Identification | 15 |  | Development Team  Remote Data Layer Team |  |  |
| DLTS63 | Bug Fixing | 60 |  | Development Team  Remote Data Layer Team  Local Data Layer Team |  |  |
| DLTS64 | Fix Report and Code Update | 30 |  | Development Team  Remote Data Layer Team  Local Data Layer Team |  |  |
| DLTS65 | Integration Testing | 14 |  | Software Quality Assurance Team |  |  |
| DLTS66 | Performance Testing | 30 |  | Software Quality Assurance Team |  |  |
| DLTS67 | Security Testing | 21 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |
| DLTS68 | Beta Release | 30 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |
| DLTS69 | Enhanced Beta | 45 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |
| DLTS70 | Final Release | 45 |  | Development  Local Data Layer Team  Remote Data Layer Team  Software Quality Assurance Team |  |  |

# References

|  |  |
| --- | --- |
| [1] | Technavio.com, 2018. [Online]. Available: https://www.technavio.com/report/global-online-language-learning-market-industry-analysis. |
| [2] | Research and Markets, "Global $8 Bn Cloud-based Language Learning Market 2019-2024: Growing Requirement of Technology-Enabled Tools that can Facilitate User Engagement, Motivation, and Collaboration," Research and Markets, 11 2 2019. [Online]. Available: https://www.prnewswire.com/news-releases/global-8-bn-cloud-based-language-learning-market-2019-2024-growing-requirement-of-technology-enabled-tools-that-can-facilitate-user-engagement-motivation-and-collaboration-300793002.html. |
| [3] | Rosetta Stone Ltd., "Rosetta Stone," [Online]. Available: https://www.rosettastone.com/. |
| [4] | Duolingo Inc., "Duolingo," [Online]. Available: https://www.duolingo.com/. |
| [5] | Rosetta Stone, "Rosetta Stone Language Tutoring Session," 3 11 2015. [Online]. Available: https://www.youtube.com/watch?v=kr\_A1fBRIAw. |
| [6] | Rosetta Stone, "Rosetta Stone Home on Facebook," [Online]. Available: https://www.facebook.com/RosettaStone/. |
| [7] | Rosetta Stone, "Rosetta Stone on Twitter," [Online]. Available: https://twitter.com/rosettastone?ref\_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor. |
| [8] | P. Allan, "Language Learning Showdown: Rosetta Stone vs. Duolingo," 2017. [Online]. Available: https://lifehacker.com/language-learning-showdown-rosetta-stone-vs-duolingo-1790938306. |
| [9] | J. Duffy, "The Best Language-Learning Software for 2019," PC Magazine, 11 4 2019. [Online]. Available: https://www.pcmag.com/roundup/261786/the-best-language-learning-software. |