

**TechnoBabel**

*Development Plan Documentation*

Chris You

Keith Zane

Justin Stryjewski

Bryan Robinson

**1.1 Project Overview**

TechnoBabel is a mobile application that offers users assistance in company specific terminology.  It is mainly targeted towards company employees that would like to learn definitions to terms that are specific to their company without the need of asking another co-worker.  Employees will have the ability to manage terminology by adding and editing its definition.  The application will also have the ability to listen to conversations, identify relevant technical terms, and display them on a device.  This will offer an easy to use dictionary of company wide acronyms and jargon that can inform new employees. It can also be a refresher to those that have been working at the company for a while.

**1.2 Project Purpose, Scope, Objective**

The goal of this project is to create a mobile application that has functionality oriented around a technical dictionary database.  The application offers users (employees) options to access/create/update terminology including attributes such as names, acronyms, descriptions, and pronunciations.

Our project will include several user functionalities such as: login credentials, searching and displaying definitions, actively listening for company specific terminology, and adding and/or editing vocabulary. This will also be supported by a database which serves as a master list for all the users and their company’s definitions.

**1.3 Team Organization (Roles and Responsibilities)**

While all of our team members will take lead on certain roles, they will have input into every aspect of the project. Thus, we have given roles to each of the members of the team. This includes the  following responsibilities:

Team Lead Keith Zane

Documentation Lead Chris You

Front-end Lead Bryan Robinson

Backend Lead Keith Zane

QA Lead Bryan Robinson

Presentation Lead Justin Stryjewski

**1.4 Problem resolution policies**

Dispute Policy

If a dispute occurs amongst team members, it is the team lead’s responsibility to resolve the dispute. If a conflict occurs with the team lead, the TA will then be contacted to come up with a resolution. If disagreement amongst team members occur regularly, a mandatory team meeting will be held to determine further steps towards preventing any escalations. If the situation escalates, the professors and TA will be contacted immediately. Disputes with our client is completely unacceptable and will result in removal from the team; however, if members are concerned about client decisions, the TA or professors will then be contacted.

Technology Policy

Before being able to choose between specific technologies, the client’s opinion will be taken into account. Taking the client’s point of view into consideration, team members must come to a consensus on the technologies being used. If team members cannot come to an agreement, both parties will suggest their choice of technology with a supporting argument ultimately leading to a final vote*.* If the conflict still persists, the client’s opinion will influence the final decision.

Attendance Policy

Team members are required to physically attend each meeting with the client or TA.  Meetings arranged by the team must be attended by teammates in person or online. If a team member cannot meet at the designated time, that individual must first let the team know in a timely manner and then be available via Google hangouts or any other convenient means of video communication. Teammates that are excused from meetings under special circumstances must be aware of all the content discussed in the meeting.

Code Quality/Deadline Policy

The quality of code has a significant impact on this project.Team members are required to submit acceptable code to the repository.  Tasks must also be fulfilled by deadlines that are established by the team.  In the case of a missed deadline, the team member must provide an explanation for the missed deadline. The group will then decide on a course of action based on the importance of the deadline. In order to keep the code organized, each member must comment their sections of code. To adhere to the project deadline policy submission, team members will submit code at least two days prior to target dates.

**1.5 Problem Plan (iteration, project schedule)**

There will be a meeting once per week with our client in person. These meetings will take place every Tuesday at 6:30pm.  Our team plans on meeting several times a week.  All meetings will be scheduled via Slack application.

The team will have scheduled iterations every week. These iterations will begin with a debriefing meeting on Monday at 7:30pm. This meeting will go over the progress that the team has made so far and what plans to be accomplished in this iteration. The iteration will end with a meeting on Saturday which will discuss everything the team has done, any concerns from any team members, and work that has not been accomplished.

This schedule is to be updated as more tasks become available and will reflect what the team is working on.

|  |  |  |
| --- | --- | --- |
|  | Week Dates | Major Tasks |
| Iteration 1 | September 11th - September 16th | * High Level Presentation * Development Plan Document   + Draft (September 16th) * Client Meeting #1 (September 14th):   + Purpose: Establish first contact, initial plans, and technologies |
| Iteration 2 | September 18th - September 23rd | * Development Plan Presentation   + Edit changes to Development Plan (September 18th)   + Create presentation slides for Development Plan (September 18th)   + Submit Development Plan Presentation and document  (September 20th)   + Development Plan Presentation (September 20th) * Client Meeting #2 (September 19th):   + Purpose: Confirm tech stack with client, requirements gathering |
| Iteration 3 | September 25th - September 30th | * Practice With Confirmed Technology * Discuss Project Architecture * Client Meeting #3 (September 26th):   + Requirements revising. |
| Iteration 4 | October 2nd - October 7th | * Lay Down Foundation For Prototype One * Client Meeting #4 (October 3rd):   + Purpose: Client Approval Of Requirements |
| Iteration 5 | October 9th - October 14th | * Requirements Presentation * Client Meeting #5 (October 10th):   + Purpose: TBD |
| Iteration 6 | October 16th - October 21st | * Client Meeting #6 (October 17th):   + Purpose: TBD |
| Iteration 7 | October 23nd - October 28 | * First Prototype Presentation * Design Specification Presentation * Client Meeting #7 (October 24th):   + Purpose: TBD |
| Iteration 8 | October 30th - November 4th | * Client Meeting #8 (October 31rd):   + Purpose: TBD |
| Iteration 9 | November 6th - November 11th | * Test Plan * Second Prototype * Client Meeting #9 (November 7th):   + Purpose: TBD |
| Iteration 10 | November 13th - November 18th | * Client Meeting #10 (November 14th):   + Purpose: TBD |
| Iteration 11 | November 20th -  November 25th | * User/Admin Manuals * Third Prototype * Client Meeting #11 (November 21st):   + Purpose: TBD |
| Iteration 12 | November 27th - December 2nd | * Client Meeting #12 (November 28th):   + Purpose: TBD |
| Iteration 13 | December 4th - December 11th | * Final Presentation * Final Prototype * Client Meeting #13 (December 5th):   + Purpose: Client Acceptance Testing |

**1.6 Configuration management**

Each team member will be assigned their individual tasks primarily based upon the technology they feel most comfortable with, and the level of interest they possess to accomplish certain tasks. Although, each team member will not be limited to one specific task of the project to maintain by themselves. Every team member will connect with each part of the project in order to distribute and collectively contribute to the project as a whole.

Our chosen sourced control method will be handled by Team Foundation Server (TFS). This method of source control is conveniently configured to work with our integrated development environment (IDE) and also used for a free private repository by request of our project client.

We will be branching our master branch based on the feature branch workflow. Whoever is assigned to a feature must branch their code when starting a new feature. After the feature is completed, the team member will present their code before merging to the the master branch. If the team approves, the branch will be merged and then tested to make sure the introduction of a new feature does not break the existing program.

**1.7 Technologies**

All technologies must comply to MIT License or other license that allow for free and legal commercial use. The possible technologies that we are looking into are as follows:

* Front End:
  + Xamarin
    - C#
  + Ionic Framework
    - JavaScript
    - HTML
    - CSS
* Back end:
  + ASP.net
    - SQL Server
    - Microsoft Azure
  + Firebase
  + Node.js