Junior Project Proposal
To The Rescue!

You saved him!

Submitted By: *Lake Sain-Thomason*
lakesainthomason@gmail.com

Matthew Del Fante
matthewdelfante@gmail.com

Robert Mooers
robert.mooers@oit.edu

Stephanie Vetter-Derosier
Stephanie.Vetter@oit.edu

Submitted To: *Todd Breedlove*
Todd.Breedlove@oit.edu

October 30, 2016
V1.1

Legal Notice

To the Rescue! makes no warranties of representation of any kind concerning the accuracy or suitability of the information contained in this document for any purpose. All such information is provided as is and with specific disclaimer of any warranties of merchantability, fitness for purpose, title and/or non-infringement. In no event shall To The Rescue!, its employees or agents be liable for any direct, indirect or consequential damages resulting from the information provided in this document. This exclusion and limitation only applies to the contrary in any written license or subscription agreement from To the Rescue! in respect of this document or references.

Copyright Notice

Revision History

Authors	Date	Version	File Name	Comments
Lake Sain-Thomason, Matthew Del Fante, Robert Mooers, Stephanie Vetter-Derosier	October 16, 2016	v1.0	Proposal.pdf	Formatted brainstormed content into a cohesive document.
Lake Sain-Thomason, Matthew Del Fante, Robert Mooers, Stephanie Vetter-Derosier	October 20, 2016	v1.1	Proposal1-1.pdf	Modified Database information in section 4.1.

Signature Page

This document accepted by:

_____   _____
Signature (Todd Breedlove)   Date

This document submitted by:

_____   _____
Signature (Lake Sain-Thomason)   Date

_____   _____
Signature (Matthew Del Fante)   Date

_____   _____
Signature (Robert Mooers)   Date

_____   _____
Signature (Stephanie Vetter-Derosier)   Date

Contents

Introduction

1.1 Purpose

The purpose of this document is to propose the design and implementation of To the Rescue!. The sections and subsections following are dedicated to define or describe the format of this document or the intended design plans. If this proposal is accepted, it will serve as a guide to constrain the implementation process.

1.2 Scope

The scope of this document is limited to project management, general system discussion, and a description of the product requirements which will describe in limited detail the intended design features and functionality. Also included is the specific functionality of modules and otherwise discrete functionalities that may later be added to the system.

1.3 Intended Audience

The audience this document is intended for includes Todd Breedlove, all members of To the Rescue!, and any third party interested or curious about this project.

Project Management

2.1 Change Management Procedures

In the event there is a change request to modify the system, the CAT Team will be informed about the change. The Change Request Form can be found in Appendix B and will provide the information outlined in Section 2.2 of this document.

2.2 CAT Team, Timelines, and Documentation

CAT Team   The CAT Team will consist of the members of To the Rescue! and Todd Breedlove. The team will evaluate the impact a change will have on the developing system. Submitted changes can be accepted or rejected by the team with a brief statement explaining the reasons for the decision. Submitted changes whether accepted or rejected will be archived in a cloud storage medium managed by the members of To the Rescue!.

Medium   Any changes must be submitted, in the format presented by the Change Request Form (Appendix B), by any member of the CAT Team. Submission is to be physical copy only digital submissions will not be accepted. A form for each CAT Team member must be submitted for the change request to be considered.

Response Time   The Change Request Form will be processed over the course of two to three business days. Responses will be emailed to the addresses detailed on the submitted Change Request Form.

Impact Analysis   Before a change moves forward, the impact that it will have on the systems development timeline must be inspected. If a change is proposed that would move the final production version of the system past the maximum allotted time, the change will be automatically denied.

Time Frame   In the event that a change is accepted, the development timeline of the system will be modified to accommodate the change. This modification is allowed to decrease or increase the development timeframe.

2.3 Software Delivery, Installation, and Acceptance Criteria

Software delivery will be distributed to all interested parties for evaluation by the end of spring term of the 2016 – 2017 school year. Included in the delivery will be a web application, written documentation of system requirements and written documentation for the startup

process. Acceptance criteria will be based on completeness and operability of the provided items and software.

2.4   Documentation and Online Help

Software documentation will be provided in written format. Documentation provided will detail the system requirements, startup process, and operation of the software described by this document.

2.5   Project Risks

Risks for this project involve the developers learning the proposed front end programming language (JavaScript) and the environment that the proposed application will run (Google Chrome). Also, the application does not have a defined amount of content required, leaving the application feeling incomplete if the developers do not fill it properly.

2.6   Customer Responsibilities

The customer will be responsible for providing their own mobile device or personal computer loaded with Google Chrome.

2.7   Status Reporting

Status reporting will be submitted weekly in memo format to Todd Breedlove. The report will include the following:

- Work completed during the current week

- Work to be completed during the next week

- Issues found during current weeks work

System General Description

3.1   Problem Statement

We will be writing a children's learning application as a progressive web application. The application will consist of numerous mini games that will align with the Oregon State Standards for Kindergarten.

The main interface will be a game map consisting of mini games along a path. The goal is to complete all mini games and rescue an animal at the end of the path. Once the user rescues an animal, the animal will be added to his or her Animal Sanctuary. The Animal Sanctuary represents the progress and achievements made by the user. The user will move to the next map after rescuing the animal. The difficulty of the mini games will increase, decrease, or remain the same depending on the user's mini game progress/success. A congratulatory splash screen will appear once all the maps have been completed. The user may play through the game as many times as he or she may like to continue to add animals to the sanctuary.

Upon launching the application, the user will have the option to create a user account, login to an existing account, or play the game without a user account. A user account will allow the user to create profiles and pick avatars to associate with these profiles. The user account tracks and saves the user's progress throughout the game. The user's animal collection will also be saved. In free play mode, the user's progress and animal collection are not saved.

Users will not be required to login to the app before they can use it, but a login will be required to create multiple profiles and to move saves between different devices. The user will be able to create an account from within the application with only an email, which will be his or her username, and a password. Passwords will be encrypted before being sent from the client to the server.

Upon logging in or creating a new account, the choose profile page will appear. From this page, the user will be able to choose to continue an existing profile or to create a new profile.

The main menu will appear once the user selects a profile, or if they choose to play without an account. From the main menu, the user may choose Play, which takes them to the game map, or the user may select Options to customize game play. In the options menu, the user may toggle sound, set the difficulty level of the mini games, and filter mini games by subject.

User information that will be stored in the database includes the user's email (username), password, profile name, profile avatar, game progress and the animals collected. References to the game map images will be stored in the database as well. Mini game information including a reference to the game code, game data and statistics for the games will be located in the database. Lastly, the database will have the game's state stored in it as well.

3.2   Perspective

3.2.1   History

The idea for To the Rescue! was a collaborative group effort that supports the abilities and interests of all group members. The To the Rescue! project also provides opportunities for all members of the team to learn by experiencing new design and programming challenges. This project requires creativity and diversity to keep the content new and engaging. The developers wanted a project that would incorporate the interests of all members. This project also promotes education and can be extended to reach users of different ages and learning levels.

3.2.2   Background

Matthew Del Fante started programming two years ago and is proficient with the object-oriented and imperative programming paradigms. Programming languages that Matt is knowledgeable on include C++, C, C#, JavaScript and QML.

Stephanie Vetter-Derosier has a background in early childhood and elementary education. She has been programming for two years and has experience in C++, C, C#, Java, and QML. Stephanie had the opportunity to intern at a data analysis company in 2016. During her internship, she gained experience in project planning and development, Java, and automated software testing.

Lake Sain-Thomason is a fourth year software engineering student. After his sophomore year at SOU, he transferred to OIT to pursue a higher education. He is familiar with many languages such as C#, C++, Java, C, and Obj-C.

Robert Mooers has three years of programming experience, having primarily used C, C++, C#, Java, Python, and QML. Robert is also proficient in graphics design, with experience in photo editing and vector graphics.

3.2.3   Prior Release

There have been no prior releases at the time this document was written.

3.3 Major Subsystems

This application is comprised of three major subsystems, the progressive web app client, the server, and the database. The progressive web app client will contain the user interface, and interact with the user. Individual games will be played within the client, and the results and statistical data will be sent to the server. The server will accept messages from clients including state, and will respond with the next action required for the client. The database will be used to hold user data and game information.

3.4 Relation of System to Existing System(s)

To the Rescue! will use libraries packaged with the ASP.NET Model View Controller framework. We intend on using open source/free images, Windows Voice Recorder for audio and open source/free audio clips.

3.5 Hardware Platform Description

Minimum Hardware Requirements for Google Chrome, the supported browser of To the Rescue!, is: Windows 7, Windows 8, Windows 10 or later and an Intel Pentium 4 processor or later that is SSE2 capable. Required internet speed for To the Rescue! is 4Mbps or higher.

3.6 Software Platform Description

To the Rescue! will be tested and intended for the Google Chrome web browser, but should be able to run on any web browser.

Product Requirements

4.1   Functional

1. Navigate to web application
    1.1. Supported on Google Chrome
    1.2. Supported on mobile devices
        1.2.1. Progressive web application
    1.3. Display Initial Splash Screen for Parents
        1.3.1. Button to Login
            1.3.1.1. Display drop down login box
        1.3.2. Button to Create Account
            1.3.2.1. Display Create Account drop down box
        1.3.3. Button to play without an Account or saving progress
            1.3.3.1. No login or account required
            1.3.3.2. Avatar chosen for user
            1.3.3.3. User unable to enter name for avatar
            1.3.3.4. User progress is not saved
            1.3.3.5. Mini-games start at easiest difficulty level
    1.4. Background music
2. Login-Drop Down Box
    2.1. Option for user to login to existing account
        2.1.1. User enters username (email) and password
            2.1.1.1. Go to Choose Profile
        2.1.2. Button to reset password
            2.1.2.1. Send a temporary password to email
            2.1.2.2. User enters temporary password
            2.1.2.3. Prompted to enter new password
3. Choose Profile Page
    3.1. List all profiles for the account
    3.2. User selects a profile
        3.2.1. Go to Main Menu
    3.3. Button to Create new profile
        3.3.1. Go to Create New Profile page
    3.4. Drop down box to delete a profile
        3.4.1. Prompt user to enter profile name and account password to delete profile
4. Create New Account-Drop Down Box
    4.1. User enters email address for username
        4.1.1. Check that username aren't already in database
    4.2. User creates a password
        4.2.1. Alpha characters and numerics only
    4.3. Redirect to Choose Profile Page
5. New Profile page

    5.1. User enters profile name

    5.2. User selects an avatar for profile pic and game character

        5.2.1. Display all available avatars

    5.3. After creating the profile, redirect to Choose Profile Page

6. User accounts

    6.1. Account information stored in database

        6.1.1. Profile name

        6.1.2. Profile avatar

        6.1.3. Email login (username)

        6.1.4. Login password

            6.1.4.1. Encrypt before transmitting to server

        6.1.5. User game progress

            6.1.5.1. Map level

            6.1.5.2. Last mini-game completed (location on path)

            6.1.5.3. Performance statistic (success percentage)

                6.1.5.3.1. Implicitly determines mini-game difficulty

            6.1.5.4. Animal collection

7. Main Menu

    7.1. Main menu appears once the user has picked a profile or is playing in free play mode

        7.1.1. Main menu background music

    7.2. User may choose to go to Animal Sanctuary

        7.2.1. Display animals that user has rescued

        7.2.2. Animals make noise and small movement when clicked

        7.2.3. Animals move around Sanctuary

        7.2.4. Button to go back to Main Menu

        7.2.5. Fun Facts listable for animals

        7.2.6. Background music

    7.3. User may choose to go to Options Menu

        7.3.1. Options menu requires password entry to change anything

        7.3.2. Button to Log Out

            7.3.2.1. Automatically default to staying logged in when the app closed

        7.3.3. User may toggle sound

        7.3.4. User may filter mini-games by subject

            7.3.4.1. Play only literacy games

            7.3.4.2. Play only mathematics games

            7.3.4.3. Play a mix of literacy and mathematics games

            7.3.4.4. Content based on Oregon State Standards/Common Core Standards (CCS) for Kindergarten

        7.3.5. User may choose to set game difficulty

            7.3.5.1. Defaults to easiest difficulty

            7.3.5.2. Button to switch/create/delete profile

            7.3.5.3. Go to Choose Profile Page

        7.3.6. Button to go back to Main Menu

    7.4. User may choose to play the game

        7.4.1. Restore users progress and mini-game difficulty if logged into an existing account

8. Game Play:Maps and Mini-Games
    8.1. Display game play Map
        8.1.1. Map contains a path with nodes as mini-game stopping points
        8.1.2. Users avatar is shown on map at current location
        8.1.3. User stops at each node along the path
    8.2. Sidebar Options Menu
        8.2.1. Button to exit map and return to Main Menu
        8.2.2. Button to go to Options
        8.2.3. Button to go to Animal Sanctuary
    8.3. Must play mini-game at current node to move to next node
        8.3.1. Must play entire mini-game to pass
            8.3.1.1. User does not pass or fail based on performance
            8.3.1.2. User praised for excellent work regardless of performance
            8.3.1.3. Mini-games reinforce correct answers and positively correct wrong answers
            8.3.1.4. Save users progress and performance statistic in database after each completed mini-game
        8.3.2. Mini-games are randomly selected for maps
            8.3.2.1. Mini-games appear more than once
                8.3.2.1.1. Different content when they do appear multiple times
            8.3.2.2. Track mini-game appearance so same game doesnt appear multiple times in a row
            8.3.2.3. Mini-game content is randomly selected from database
    8.4. At end of the map is an animal to rescue
        8.4.1. User rescues animal after completing all mini-games on map
        8.4.2. The rescued animal is moved to users Animal Sanctuary
    8.5. Once a map is completed, user moves to the next map
        8.5.1. Algorithm to determine if mini-game difficulty should increase, decrease, or remain the same based on users ability to complete mini-games on previous map
    8.6. Display a congratulatory splash screen when all maps are completed
        8.6.1. Return to Main Menu
            8.6.1.1. User may play game again from beginning
            8.6.1.2. Animals are preserved in the Sanctuary
9. Database
    9.1. Create table for user accounts
        9.1.1. Email(username)
        9.1.2. Encrypt Password
    9.2. Create table for user profiles
        9.2.1. Reference to account
        9.2.2. Profile Name
        9.2.3. Avatar ID
    9.3. Create table of animal collections (for each profile)
        9.3.1. Animal ID
        9.3.2. Profile ID
    9.4. Create tables for image references
        9.4.1. Available avatars

9.4.2. Game maps
9.4.3. Animals to rescue
9.5. Create table for mini-game content(data)
    9.5.1. Reference to the location of the mini game code
    9.5.2. General game data such as:
        9.5.2.1. Colors, words and reference to images
        9.5.2.2. Math sequences
        9.5.2.3. Sight words and letters
        9.5.2.4. Numbers and corresponding words
9.6. Create user progress table (stores upon mini-game completion)
    9.6.1. Stores current map
    9.6.2. Stores location (node) on map
    9.6.3. Stores the ID of the last three mini-games completed

4.2   Performance

To the Rescue! will run at 60 frames per second to provide smooth transitions between pages and silky movements of small characters. Older devices may experience a lower or choppy framerate that do not fit the hardware specifications.

4.3   Reliability

To the Rescue will try to maintain 95% reliability for up to 30 concurrent users. It is important that users are able to rely on the application functioning properly, but the server may unexpectedly go down. Thus, 95% reliability is reasonable. If more than 30 users use the application concurrently, the application will become less reliable.

4.4   Data Description

To the Rescue! will be serving web pages, including images, audio, and JavaScript. Data will be transferred from the database and to the database through an internet connection. The transfer rate will be 4 Mbps with a maximum page load time of 2 seconds.

4.5   Security and Safety

All passwords will be encrypted before being sent off to the database, and users are logged out when the application is closed.

4.6   Constraints

'To The Rescue!' content will be based on the Oregon State Common Core Standards for Kindergarten.

User Profiles

The users of this software will be parents who create accounts for their children and the children use the application. The target users are children at a preschool and kindergarten learning level.

A - Glossary of Terms

Progressive Web App A new way to create web applications that is responsive to a user's mobile device. Progressive Web Apps allows users to add the app to the home screen of the user's mobile device with one click, and does not require installation.

B - Change Request Form

Name (Print)	Email Address	Date

Change Request:

Purpose for Change Request:

Additional Comments: