2020

2021

A close up of a logo

Description automatically generated

Company: DNASC (Definitely Not A Software Company)

Submitted to: Calvin Caldwell

Submitted by: Jihyuk Chung

Version: 2.0

Email: jihyuk.chung@oit.edu

Date: June 08, 2020

LEGAL NOTICE

All product, support, services, information, and software on this site are provided “as is” without warranty of any kind, express or implied, including, but not limited to the implied warranties of fitness for a particular purpose, and non-infringement. In no event shall DNASC or other contributors be liable to you or any third party for any direct or indirect, special, incidental, or consequential damages in connection with or arising out errors, omissions, delays or other cause of action that may be attributed to your use of any product, support, services, information or software from this site, including, but not limited to, lost profits or lost data, even if DNASC or other contributors had been advised of the possibility of such damages.

COPYRIGHT NOTICE

**Copyright © 2020-2021 DNASC. All rights reserved.**

All original work including documents, questions, references, images, program, source code or other materials whatsoever contained herein, or obtained through the DNASC website, mobile application, or server are protected by copyright. Except as explicitly permitted below, these materials may not be reproduced in whole or in part, in any form or by any means, including photocopy, electronic storage and retrieval, or translation into any other language without the express written consent of the copyright holders.

REVISION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authors | Company | Version | Date | Comments |
| Jihyuk Chung | DNASC | 1.0 | April 18, 2020 | Rough Draft |
| Jihyuk Chung | DNASC | 2.0 | April 18, 2020 | Final Draft |
|  |  |  |  |  |

SIGNATORY PAGE

This document is accepted by:

|  |  |
| --- | --- |
| X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_ |
| Author: Jihyuk Chung | Date |
|  |  |
| X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_ |
| Reviewer: Calvin Caldwell | Date |

TABLE OF CONTENTS

[LEGAL NOTICE i](#_Toc42259550)

[COPYRIGHT NOTICE i](#_Toc42259551)

[REVISION HISTORY i](#_Toc42259552)

[SIGNATORY PAGE ii](#_Toc42259553)

[TABLE OF CONTENTS iii](#_Toc42259554)

[1. INTRODUCTION 1](#_Toc42259555)

[1.1. PURPOSE 1](#_Toc42259556)

[1.2. SCOPE 1](#_Toc42259557)

[1.3. INTENDED AUDIENCE 1](#_Toc42259558)

[2. PROJECT MANAGEMENT 2](#_Toc42259559)

[2.1. CHANGE MANAGEMENT PROCEDURES 2](#_Toc42259560)

[2.2. DOCUMENTATION 3](#_Toc42259561)

[2.3. CUSTOMER RESPONSIBILITY 3](#_Toc42259562)

[2.4. SOFTWARE ACQUISITION 3](#_Toc42259563)

[2.5. PROJECT RISKS 3](#_Toc42259564)

[2.6. STATUS REPORTING 3](#_Toc42259565)

[3. SYSTEM GENERAL REQUIREMENTS 4](#_Toc42259566)

[3.1. PROJECT SUMMERY 4](#_Toc42259567)

[3.2. RELATION OF SYSTEMS TO EXISTING SYSTEM 5](#_Toc42259568)

[3.3. HARDWARE PLATFORM DESCRIPTION 5](#_Toc42259569)

[3.4. SOFTWARE PLATFORM DESCRIPTION 6](#_Toc42259570)

[3.5. THIRD PARTY LIBRARIES 6](#_Toc42259571)

[4. PROJECT REQUIREMENTS 7](#_Toc42259572)

[4.1. FUNCTIONAL 7](#_Toc42259573)

[4.2. PERFORMANCE 9](#_Toc42259574)

[4.3. DATA TRANSFER DESCRIPTION 9](#_Toc42259575)

[4.4. SECURITY / SAFETY / CONSTRAINTS 9](#_Toc42259576)

[5. USER PROFILE 10](#_Toc42259577)

[6. APPENDIX A – GLOSSARY OF TERMS 11](#_Toc42259578)

[7. APPENDIX B – CHANGE REQUEST FORM 12](#_Toc42259579)

1. INTRODUCTION
   1. **PURPOSE**

The purpose of this document is to propose the design and implementation strategy for this project, Cato. The rest of the document is responsible for defining the document’s format and the proposed project design. Upon acceptance, this proposal will be a living document which will guide the development and implementation and maintenance process.

* 1. **SCOPE**

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

* 1. **INTENDED AUDIENCE**

This document is intended for Calvin Caldwell and any other interested party not listed.

1. PROJECT MANAGEMENT
   1. **CHANGE MANAGEMENT PROCEDURES**
      1. **The Change Administration Team**

Should a change to this product become necessary, the Change Administration Team, (CAT), will convene and discuss the requested change. The CAT team will consist of Jihyuk Chung, Calvin Caldwell and all of whom will consider the ramifications of the requested change to the product. Any requested changes are subject to rejection. The change request form can be located within Appendix B and will contain all relevant details from the following section.

* + 1. **Medium**

Any changes must be submitted in the format presented by the Change Request Form (Appendix B) to a member of the CAT team.

* + 1. **Response Time Protocol**

All change request will be processed within a maximum of 7 business days. Upon acceptance, a member of the CAT team will notify the requester of the resulting decision via the email provided on the Change Request Form.

* + 1. **Impact Analysis**

Before a change is accepted, the impact that it will have on the system’s development timeline must be analyzed. If the suggested change causes the project to go significantly over or under its scope, the Change Request Form will be denied.

* + 1. **Time Frame**

If a Change Request Form is accepted, future sprints must be analyzed and modified to accommodate the new change.

* + 1. **Archive**

All changes, whether accepted or denied, must be archived in the Revision History section of this document (see page i for reference).

* 1. **DOCUMENTATION**

All documentation for the product will be provided alongside the website and app at the dedicated Github page.

* 1. **CUSTOMER RESPONSIBILITY**

The customer will be responsible for providing their browser and android devices and access to an internet connection.

* 1. **SOFTWARE ACQUISITION**

The latest version of the product will be available at the GitHub address, throughout the product’s development.

* 1. **PROJECT RISKS**

The risks associated with this project include the following: learning a new programming concepts (AI and BERT methods), finding suitable APIs, managing time, managing workloads, and navigating the agile workflow.

* 1. **STATUS REPORTING**

Once per week, Jihyuk Chung will send status reports to or meet with Calvin Caldwell, on a day agreed upon by all members of the CAT team. The status report will include a list of work which has moved to complete from the previous week. If any issues or risks have been identified in the previous week, these will also be included in the status report and discussed to mitigate or remove them.

1. SYSTEM GENERAL REQUIREMENTS
   1. **PROJECT SUMMERY**

Cato is an artificial intelligence chatbot for a CST 116, 126, and 136 C++ classes. This project aims to be a student helper tool and automate interactions between students and professors. The tool will have an AI agent fluently answers questions on multiple platforms including a website and android app. The frontend platforms include three main sections, the login screen, the main program, user profile, and settings. The main program is going to contain the chat system with the chatbot and chat history based on users. The website and app will be designed to have modern, fast, and responsible service. Setting the AI and multi-platform capability will involve implementing Tensorflow, TFLearn, NLTK, and Firestore database.

Prioritizing the app, the main screen will be different for students and professors. For students, they can interact with the chatbot and see previous chat activities. The chatbot will answer general questions and questions related to the study materials. It is necessary to hardcode intents with questions and answers. However, the chatbot will interpret similar questions and find patterns to identify the right answers. For example, “how pointers work”, “what is a pointer”, “give an example of pointers in C++” are not the same question but shares the similar pattern and answers. When the chatbot is unable to identify the pattern, it should inform that to the users instead of picking a random answer from the database. The unanswered questions will be saved and send to the professor if the student accepts it. For professors, they can not only interact with the chatbot for testing but also be able to respond to unanswered questions given as described above. When the professor answers the question, the program will update the intent. By this automated process, the program will continuously grow as more questions answered. The program will aim to have enough database to help students fluently and have human-like interactions. A late goal of this project is to use pre-trained BERT (Bidirectional Encoder Representations from Transformers) method and figure out answers just from given study materials by itself.

The login screen, user profile, and settings will give secure and user-based experience to users. They will be able to access the user specific information in different platforms and devices using their accounts. The account based on emails helps to block non-authorized users if not wanted. This can be helpful to keep the app OIT specific and avoid sending too many unanswered questions to professors. In this case, either registering or sending unanswered questions will be limited to OIT emails only.

The program will involve implementing three AI related frameworks and one database. Tensorflow (or similar frameworks such as PyTorch and Dialogflow) is an open source artificial intelligence library. TFLearn is a deep learning library API for Tensorflow and will be used to identify patterns from the questions. NLTK is a natural language toolkit to help AI understand user input. These frameworks imposed to build an AI not from the scratch. Also, Firestore database will be used to keep the program real-time and have short time offline capabilities for the mobile app.

* 1. **RELATION OF SYSTEMS TO EXISTING SYSTEM**
     1. **Website Hardware Platform**

N/A

* 1. **HARDWARE PLATFORM DESCRIPTION**
     1. **Website Hardware Platform**

For Windows, Windows 7, Windows 8, Windows 8.1, Windows 10 or later

For Mac, OS X Yosemite 10.10 or later

For Linux, 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

A device with internet connection.

An Intel Pentium 4 processor or later that's SSE2 capable

4GB of RAM and 100 MB of free memory.

* + 1. **Mobile App Hardware Platform**

A mobile device with Android 6.0 (API 23 M) or above.

Internet connection and Google Play Store accessibility.

4GB of RAM and 100 MB of free memory.

* 1. **SOFTWARE PLATFORM DESCRIPTION**
     1. **Website Hardware Platform**

The website is supported by latest web browsers including Chrome, Firefox, Safari, and Microsoft Edge. To enhance the experience, following browser permissions may be required: JavaScript, Flash, Image, Sound, and Notifications.

* + 1. **Mobile App Hardware Platform**

The application will be downloadable from Google Play Store.

* 1. **THIRD PARTY LIBRARIES**

The third-party libraries include Tensorflow, TFLearn, NLTK, and Firestore database.

1. PROJECT REQUIREMENTS
   1. **FUNCTIONAL**
      1. **Platforms**
         1. Application written in Java and Android Studio development
         2. Website written in HTML and CSS
         3. Server Code written in JavaScript and Node.js development
      2. **Accessibility**
         1. Users shall be able to download the android application from Google Play Store.
         2. Users shall be able to visit the website through web browsers.
      3. **Login & Authentication**
         1. Users must be able to register and login to the application
         2. Registering a new account must be restricted to OIT email addresses only
         3. Users must verify their email addresses before accessing the modules
         4. Users must be able to save login information to cache
         5. User authentication will include following information
            1. Identifier (email address)
            2. Account Providers
            3. Account created date
            4. Last login date
            5. UID
      4. **User Roles & Access**
         1. User accounts will be either student or professor mode
         2. Student accounts must be redirected to the student mode
         3. The student mode will include following navigation menu:
            1. Profile
            2. Chatbot
            3. Chat History
            4. Course Materials
            5. Settings
         4. Professor accounts must be redirected to the professor mode
         5. The professor mode will include following navigation menu:
            1. Profile
            2. Chatbot
            3. Chat History
            4. Unanswered Questions
            5. Course Materials
            6. Settings
      5. **Chatbot & Chat History**
         1. Course materials must be categorized by the classes.
         2. Users must be able to ask questions to the chatbot
         3. All questions must be answered based on the AI algorithm
         4. When the chatbot cannot find a solution
            1. It must notify users that a solution is not found
            2. Send the question to professors in request of the users
            3. When questions without a solution are answered, its intent will be added to the database. The chatbot will answer newly added questions
         5. Users must be able to read and delete their own chat history
         6. Professors must be able to read and answer unanswered questions from users
      6. **Course Materials**
         1. Course materials must be categorized by the classes.
         2. Users must be able to read and download course materials
         3. Professors must be able to add course materials
         4. The chatbot must interpret the course materials using the pre-trained BERT method and use them to answer questions
      7. **User Profile**
         1. Users must be able to view and edit their profile information
         2. User database will include following information
            1. Display Name
            2. Email Address
            3. Bio
      8. **Settings**
         1. The system will have an option to view user profile
         2. The system will have an option to logout
         3. The system will have an option to delete their account
   2. **PERFORMANCE**

The website and application will move fluidly in capped fps (frames per seconds) of client devices up to 60 fps. The host will load pages without prior caching in less than 1 second. Any required delays will provide appropriate messages or animations to users.

* 1. **DATA TRANSFER DESCRIPTION**

Data will transfer at minimum of 10 Mbps from the user devices to the server and 5 Mbps from the server to users.

* 1. **SECURITY / SAFETY / CONSTRAINTS**

All passwords will be hashed internally by Firebase Authentication. When an account is uploaded with a password using a different algorithm, the password will be rehashed the first time that account logs in. Custom security rules will provide access control and data validation.

1. USER PROFILE

This application has two core user profiles. First are OIT students in CST 116, 126, and 136 C++ classes who are interested in exploring new technologies and getting additional assist from an AI chatbot. Second are professors of the classes who will answer and review questions from the students.

1. APPENDIX A – GLOSSARY OF TERMS

N/A

1. APPENDIX B – CHANGE REQUEST FORM

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Requested: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Description of Proposed Change:

Reason for change:

Comments: