Computer Science Department

Capstone Project Proposal

***Personal Assistant***

CSC 520

December 2021

By

*Brendan Geary*

**Contents**

Student Objectives 3

Problem Specification 4

Solution Processes and Design 5

Requirements Collection and Analysis 5

Use Case Diagram (UCD) 8

System Context Diagram (SCD) 9

Functional Decomposition Diagram (DFD) 10

Sequence Diagram 11

UI Mockups 12

CSC521 Supervisor Agreement 14

Benchmark Specification 16

Tool List 17

Time Schedule for CSC521 18

Grading Scheme for CSC521 19

Deliverables 20

References 21

**Student Objectives**

My student objectives for the project is to gain experience in the full lifecycle of a mobile application, this encompasses the project’s inception to its delivery to the intended audience. The main purpose however of the project is to gain experience in the actual development of a mobile application using Android Studio as well as Java with special focus on the development around integrating various API’s together, these API’s can range from the platform’s own API to other application’s own API’s. Another area I wish to gain more knowledge of as a result of this project is in NLP with learning about how to transform human speech into text and ultimately into actionable tasks for an application to perform.

**Problem Specification**

My project proposal is the creation of a Personal Assistant Application for Android devices. This assistant would be capable of performing basic functionalities at the request of the user through voice commands such as setting timers, alarms, searching the internet for information regarding a certain question, as well as other functions. It would do so by using the phone’s built-in microphone to detect when a person is speaking to it through a predetermined phrase followed by the request. This recorded audio would then be turned into text for the application to then process and understand what the user has asked. From there, depending on the circumstance the application would then perform a corresponding action in accordance with the request and display the results for the user to see on the screen. This would require the application to work hand in hand with multiple different applications that are pre-installed on most modern day phones such as their clock app, google maps, or their internet browser.

To keep user’s privacy secure, the application would be self-sufficient and entirely contained within the user’s device, forgoing the need to query an outside source for assistance in understanding the user’s request. This also means that any data collected from the user that’s necessary for the application to work would stay locally on the user’s device allowing them complete control over their own information.

The application would also need a form of memory, in which it is able to “remember” certain pieces of information such as its own name, the user’s name, and any other relevant information which would also be transparent to the user.

**Solution Processes and Design**

**Requirements Collection and Analysis**

1. **Purpose**

The expectation for the applications is the creation of a personal assistant application for mobile devices that utilizes the device’s microphone along with speech-to-text and text-to-speech capabilities.

1. **Intended Audience**

Expected Stakeholders:

Users: The ones who will be interacting with the personal assistant application on a day-to-day basis

Google: The ones who will be enabling interactivity between apps as and their search engine with respect to their own guidelines

Android: The ones who will be enabling interactivity between apps as well as the phone’s own in-built applications within their own guidelines

1. **Intended Use**

The application will serve as a personal assistant AI capable of performing tasks asked by the user such as setting timers, alarms, reminders, ect. through the use of the Android device’s first party applications.

1. **User Stories**

| **User** | **User Acuities** | **Activities Related Features** |
| --- | --- | --- |
| User | User speaks phrase  “ *x* **an alarm** for *y*” | Assistant interfaces with the phone’s in-built alarm application |
| “” | User speaks phrase  “*x* **a timer** for *y*” | Assistant interfaces with the phone’s in-built timer application |
| “” | User speaks phrase  “*x* **a reminder** for *y*” | Assistant interfaces with the phone’s in-built calendar application |
| For Above cases,  *x* can be one of the following keywords  “create”, “set”, “start”, “begin”, “initiate”, “make”  **Bold words** are the action to be undergone  *y* is a date/time format dependant on the context of the preceding action | | |
| “” | User speaks a phrase in the form of a question | Assistant searches the internet using the phone’s default browser and displays the results |
| For the above case, questions can begin with the 8 preceding keywords,  “who”, “what”, “where”, “when”, “why”, “how”, “is”, “are”  Alternatively if the user’s input does not map to any other action, the system will default to searching the input online | | |
| “” | User speaks phrase  “Open/Start/run x” | Assistant searches the phone for the specified application and starts it |
| For the above case, x is the name of application installed on the user’s device, if no such app exists user will be redirected to their default app store’s search results for the application name | | |
| “” | User speaks phrase  “Increase/decrease brightness” | Assistant adjusts the system’s settings to increase/decrease the brightness of the device |
| “” | User speaks phrase  “Set/change assistants’/user’s name to *x*” | Assistant changes the name it responds to in accordance with the given name |

1. **Acceptance criteria**

The accepted criteria for any action the user can take is for the assistant to be able to correctly interface with the required external programs in such a way to achieve the desired request made by the user.

1. **Scope**

The main goal of the project is the creation of an automated assistant that, using the users’ devices’ existing functionalities, is capable of streamlining the creation and execution of tasks that a user may want to do.

1. **Nonfunctional Requirements:**

* **Functional requirements:**
  + Security requirements:
    - Sensitive information must remain local to the user’s device
      * This could include search history, task log, etc.
    - Any sensitive information must be stored using a form of encryption
      * Applicable to autofill information used by internet applications
  + Practical requirements:
    - Able to manage user needs (timers, alarms, calendar, ect.) through standard first party apps on device
    - Ability to open and close applications on the user’s behalf when requested
    - Capable of adjusting certain system settings when asked
    - Able to utilize search engines on behalf of the user to accomplish tasks
  + Extra requirements:
    - Customization options for the application on a user-to-user level
      * Users can customize the assistant’s name
      * Users can customize their own name
      * Users can customize the voice of the assistant
      * Users can customize the application’s colors
* **Nonfunctional requirements:**
  + Security
    - Encryption standards such as SHA256 would be used to encrypt the user's private data.
    - Full transparency would be available for the user, allowing them full access to the type of information kept by the application.
  + Accessibility
    - By using Android permissions for folders, temporary storage for the application can be saved to a location wherein the user has full access to the contents.
  + Performance
    - Performance can be reasonably guaranteed due to the fact that Android applications are predominantly created using Java and therefore the device is already geared to running java applications
  + Reliability
    - Reliability can be assured through the use of accredited API’s from big name tech companies, ensuring that the standards of performance fall in line with industry standards.

1. **Assumptions and Dependencies**

This software depends on Android’s default applications to be activated on the users’ phone, otherwise the application would not be able to function as the middleman between the user and the desired task. The same can be said with Google’s functionalities within the assistant such as web search, mapping, etc.

1. **Definitions and Acronyms**

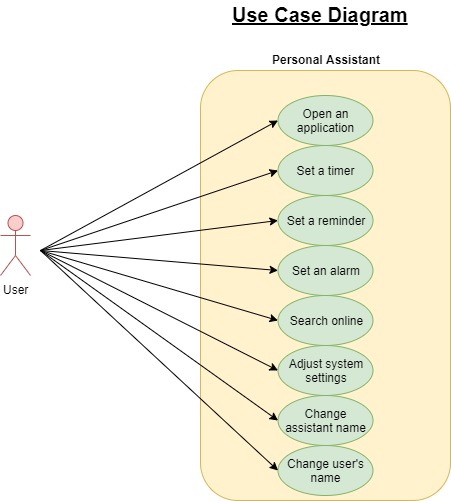
AI: Artificial Intelligence

API: Application Programming Interface

NLP: Natural Language Processing

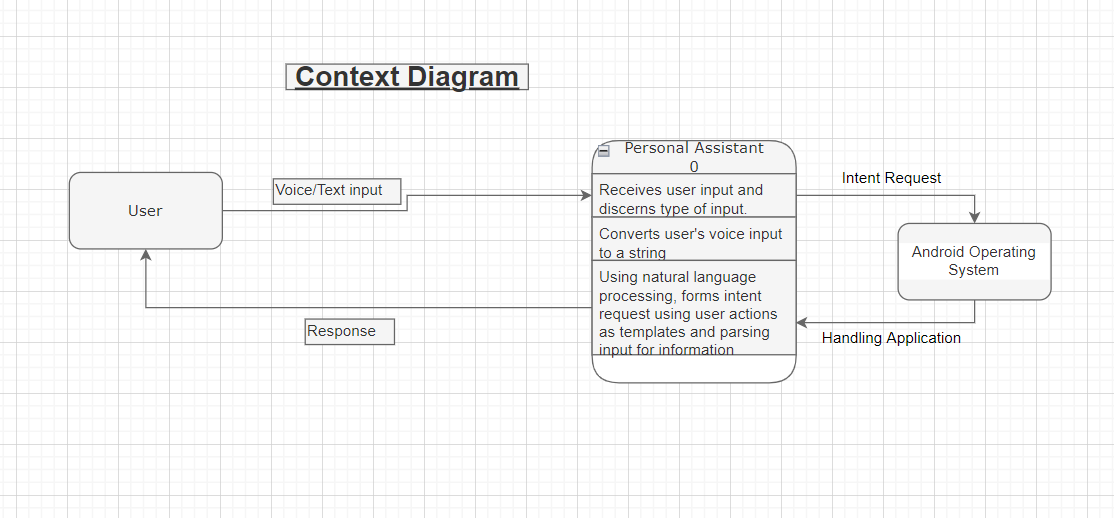
Intent: A class that

**Use Case Diagram (UCD)**

****

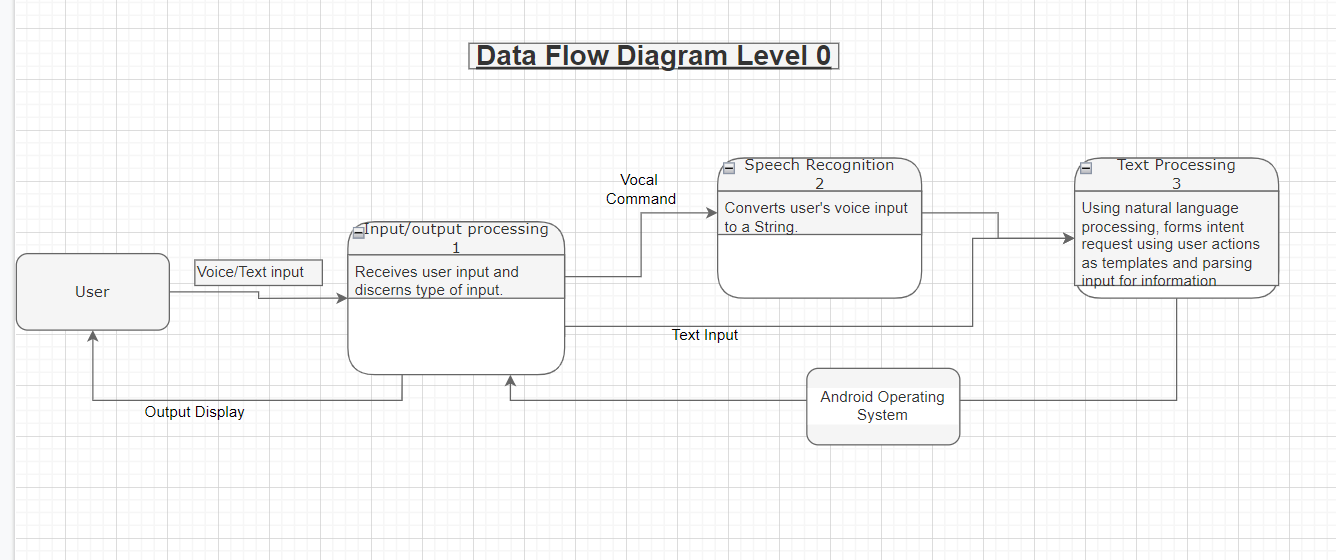
The preceding diagram shows the various actions that a user can accomplish while using the proposed application.

**System Context Diagram (SCD)**

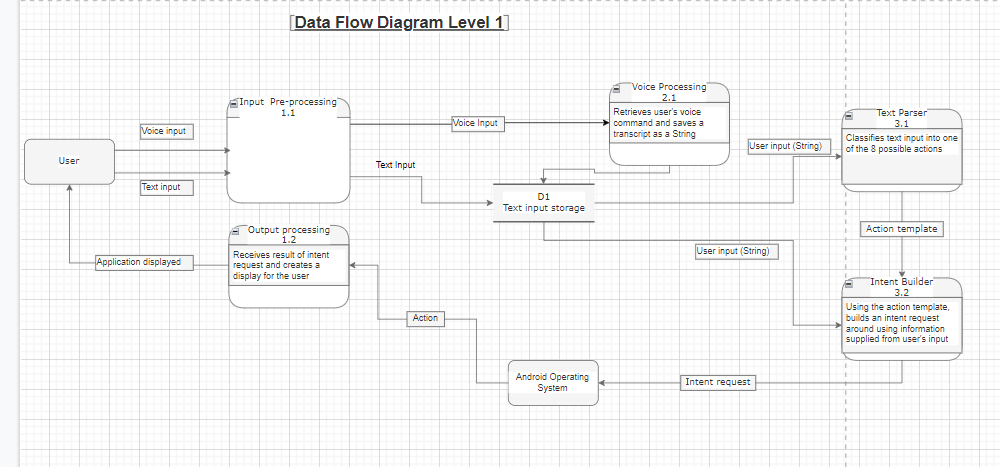
****

This context diagram shows the way that necessary parts of the program interact with each other, the user will supply a form of textual or vocal input to the program, which would extrapolate the intended action, sending a request to the Android OS to which returns the handling application.

**Functional Decomposition Diagram (DFD)**

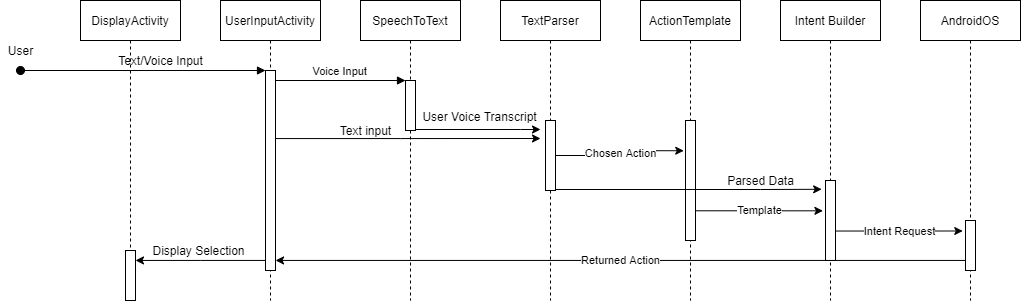
****

At level 0, the diagram represents how data flows between the different sections of the program, these sections being Input/Output processing, Speech Recognition, and Text Processing. The Input/Output Processing handles receiving the input from the User as well as displaying the handler application returned by the operating system. The speech recognition section deals with the conversion of user voice commands to a stringified representation.

****

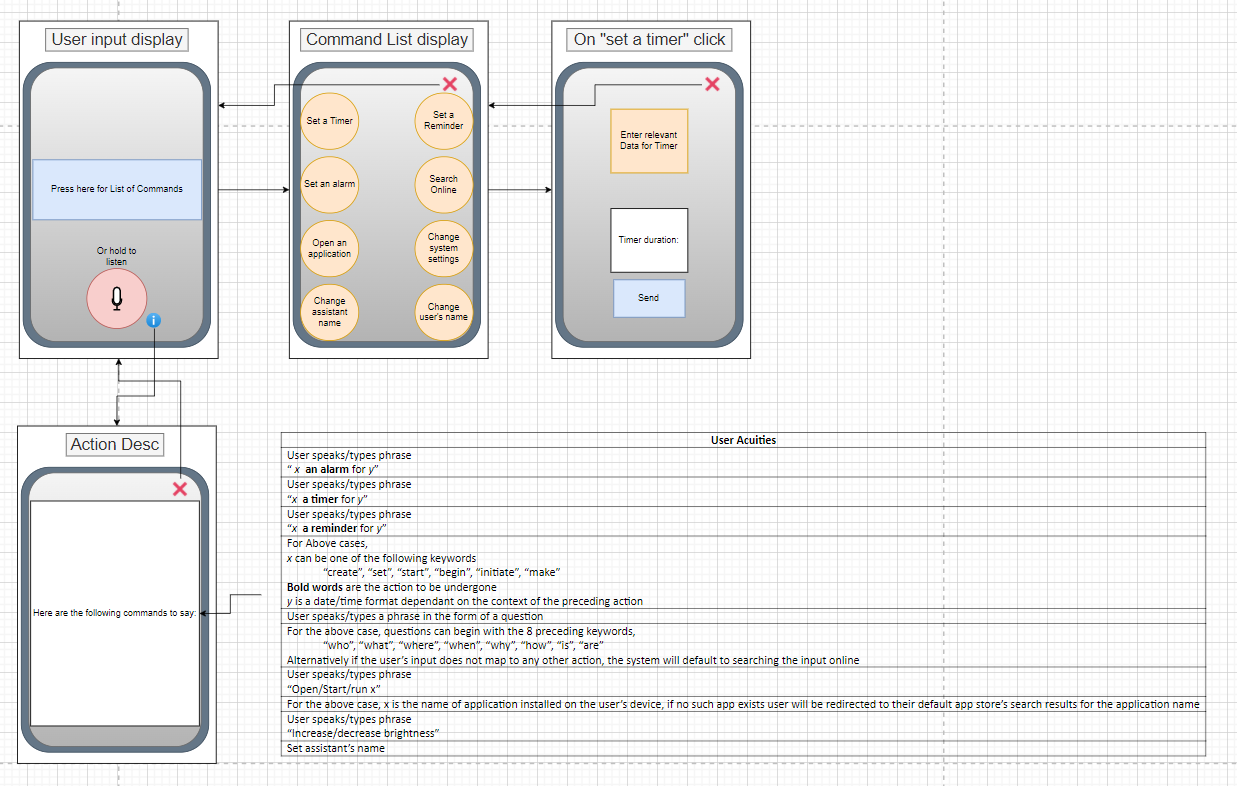
Level 1 shows a more detailed breakdown of the sections, dividing the sections into separate functions which will be used to demonstrate the class layouts.

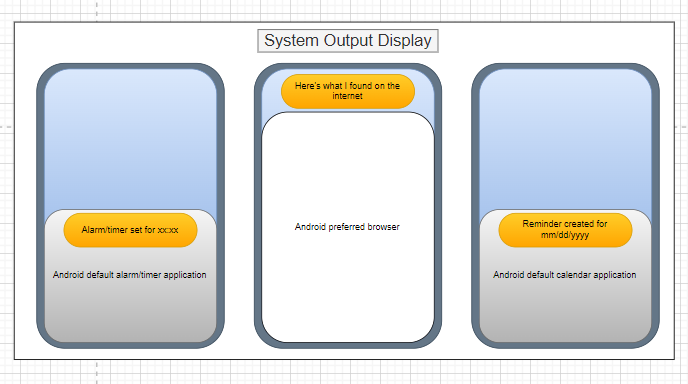
**Sequence Diagram**

****

This diagram shows the order in which the program will call the various functions to perform a task. As well as this, it gives a basic overview of the type of information that will be shared between functions.

**UI Mockups**

****

****

The above two diagrams show the layout that will be utilized by the program to convey information to the user. The user input will allow for the user to either hold the microphone button to give verbal commands to the assistant or they can go through the submenus within the commands list to enter the required data for a specified task via text entry. The small information button next to the microphone would then lead to a manual describing the various functionalities of the assistant and how to properly declare them. The System Output Display section then shows the various types of resulting screens that may be displayed to the user upon successful completion of a task.

**CSC521 Supervisor Agreement**

**SC 520**

**Supervisor Selection for CSC 521**

**Student – Supervisor Agreement**

| I / We | Brendan Geary | | would like to request | |
| --- | --- | --- | --- | --- |
|  | |
| Faculty | Allan Brockenbrough | | | |
| to supervise my/our following capstone project: | | | | |
| Project Title | Personal Assistant Application | | | |
| Problem Description | My project proposal is the creation of a Personal Assistant Application for Android devices. This assistant would be capable of performing basic functionalities at the request of the user through voice commands such as setting timers, alarms, planning routes through a mapping program, searching the internet for information regarding a certain question, as well as other functions. It would do so by using the phone’s built-in microphone to detect when a person is speaking to it through a predetermined phrase followed by the request. This recorded audio would then be turned into text for the application to then process and understand what the user has asked. From there, depending on the circumstance the application would then perform a corresponding action in accordance with the request and display the results for the user to see on the screen. This would require the application to work hand in hand with multiple different applications that are pre-installed on most modern day phones such as their clock app, google maps, or their internet browser.  To keep user’s privacy secure, the application would be self-sufficient and entirely contained within the user’s device, forgoing the need to query an outside source for assistance in understanding the user’s request. This also means that any data collected from the user that’s necessary for the application to work would stay locally on the user’s device allowing them complete control over their own information.  The application would also need a form of memory, in which it is able to “remember” certain pieces of information such as its own name, the user’s name, and any other relevant information which would also be transparent to the user. | | | |
| **Signatures Section** | | | | |
| Student(s)  Signature(s): | | Brendan Geary | | November 19, 2021 |
|  | | Click or tap to enter a date. |
| CSC521 Supervisor signature: | |  | | November 19, 2021 |

**Benchmark Specification**

In order to implement the Personal Assistant application, 6 benchmarks were created as follows with each benchmark following the previous in sequential order:

1. **Creation of the Application’s landing page**
   1. Creation of the GUI
      1. Capable of accepting user’s text input
      2. Capable of accepting user’s voice input
         1. Saving of the above to temporary storage within the user’s device
2. **Implementation of Voice Processing Module**
   1. Capable of accessing the user’s voice input from the landing page and processing it into a String
3. **Implementation of Text Processing Module**
   1. Capable of accessing the user’s text input from the Landing page’s temporary storage
   2. Capable of accessing the user’s Stringified voice input from the Speech Processing Module’s temporary storage
   3. Classify the input String according the possible actions
4. **Implementation of Intent Request Module**
   1. Capable of creating requests for Android OS to interpret
   2. Capable of returning the application given by the Android OS
5. **Implementation of Output Processing Module**
   1. Creation of individual output screens for the various user activities
      1. Capable of displaying the screens with the information extracted from the user’s input
6. **Integration Testing between modules**

**Tool List**

In order to create the application, the following tools will be used during implementation; however it may not be representative of the final tool list after completion of the application.

**Project Management Tool: Trello**

* https://trello.com/b/Ziynu6cy/geary-final-project-csc520-521

**Source Control Tool: GitHub**

* [solowraith/Personal-Assistant (github.com)](https://github.com/solowraith/Personal-Assistant)
* Show proof that your CSC521 supervisor has joined your CSC521 GitHub repository

**IDE:**

* Android Studio

**Language:**

* Java
* Python

**Libraries:**

* [GitHub - kartikn27/nlp-question-detection: Given a sentence, predict if the sentence is a question or not](https://github.com/kartikn27/nlp-question-detection)
* [GitHub - cmusphinx/sphinx4: Pure Java speech recognition library](https://github.com/cmusphinx/sphinx4)
* [15.ai: Natural TTS with minimal viable data](https://finalfinaltest.15.ai/)

**Time Schedule for CSC521**

The following is a tentative schedule for the completion of the benchmarks in order to deliver the proposed application in a timely manner. Dates and times may be subject to change during actual development of the application.

* **January**
  + **(January 02-15) Creation of UI landing page for receiving user’s request**
  + **(January 16-27) Creation of VoiceProcessing module to convert user’s voice to a String**
  + **(January 28-31) Integration testing of UI and VoiceProcessing**
* **February**
  + **(February 01-21) Creation of TextProcessing module**
    - **(Feb 01-07) Design method to receive transcript string and prepare it for NLP module**
    - **(Feb 08-21) Implementation of NLP API to detect user’s desired action**
  + **(February 21-28) Creation of ActionTemplate module to provide a standardized way of assembling required data**
  + **(February 29-31)Integration testing of TextProcessing with the UI and voiceProcessing modules**
* **March**
  + **(March 01-18) Creation of IntentBuilder method which will construct an intent request using the user’s String and ActionTemplate’s template, receives activity from AndroidOS**
  + **(March 19-21) Integration testing of IntentBuilder and AndroidOS**
  + **(March 22- April 09) Creation of Output Processing module** 
    - **(March 22-31) Design displays for each possible action requested by the user**
* **April**
  + **(April 01-09) Add functionality to choose which display to show dependant on the intent request**
  + **(April 10-14) Integration testing of all modules combined**
  + **(April 15-30) Work on final presentation**
* **May**
  + **(May 01-05) Practice project presentation**
  + **(May 05) Present project**

**Grading Scheme for CSC521**

I propose that the following weighted grade distribution will be used for CSC521 which is agreed upon with the faculty supervisor of the project.

| **UI Landing Page:** | **10%** |
| --- | --- |
| **VoiceProcessing Module:** | **20%** |
| **TextProcessing Module:** | **25%** |
| **ActionTemplate:** | **05%** |
| **IntentBuilder:** | **10%** |
| **OutputDisplay Module:** | **20%** |
| **Project Presentation:** | **10%** |
| **Total:** | **100%** |

**Deliverables**

Upon completion of the project the following items will be delivered

* Original proposal and presentation file(s) (from CSC 520)
* Amendments to the proposal (approved by the CSC521 project supervisor)
* System architecture diagram(s) (UML, DFD context, etc.), enhanced with details determined during implementation
* Appropriately commented source code
* Documentation of project functionality (test results, screenshots, video capture of project execution, etc.)
* Sample output (screen shots and/or reports)
* Executables and/or projects
* Presentation documents (used to support the presentation of the completed CSC 521 project), including any presentation file(s)
* Project journal: a narrative of the progress of the project, in clear, concise English, including any problems encountered and how said problems were addressed
* Project *postmortem*: a summary of what was learned from the project and (based on that experience) discussion of how various aspects of the project might have been approached differently
* A list of what areas of the proposal (if any) were not completed, including reasons why
* Demonstration of the completed project (PowerPoint format), including screenshots of the functioning project

**References**

1. Cmusphinx. “CMUSPHINX/Sphinx4: Pure Java Speech Recognition Library.” *GitHub*, https://github.com/cmusphinx/sphinx4.
2. kartikn27. “Kartikn27/NLP-Question-Detection: Given a Sentence, Predict If the Sentence Is a Question or Not.” *GitHub*, https://github.com/kartikn27/nlp-question-detection.
3. “Natural TTS with Minimal Viable Data.” *15.Ai*, https://finalfinaltest.15.ai/.
4. “Android Developers.” *Android Developers*, https://developer.android.com/.