



Programmer's Guide

STeMS Backend Services

Version 2.0

SWEN670

Summer 2023

Dr. Mir Assadullah

Contributors

David Babers, Mohamed Ben Lakbir, Robson De Souza, Collin Hicks

Johnny Huynh, Benny Iko, Scott McCrillis, Jonathan Nagy, Amol Thomare

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Revision History

Date	Version	Description	Author(s)
7/22/2023	1.0	Milestone 3 Version	David Babers, Mohamed Ben Lakbir, Robson De Souza, Collin Hicks, Johnny Huynh, Benny Iko, Scott McCrillis, Jonathan Nagy, Amol Thomare
7/24/2023	2.0	Milestone 4 Version	Team B Members

The following project documentation will be included in the entire documentation package for this project/application:

	Document	Version	Date
1	Project Management Plan (PMP)	4.0	8/04/2023
2	Software Requirements Specification (SRS)	4.0	7/24/2023
3	Technical Design Document (TDD)	3.0	7/24/2023
4	Programmers Guide (PG)	2.0	7/24/2023
5	Deployment and Operations (DevOps)	2.0	8/04/2023
6	User Guide (UG)	1.0	8/05/2023
7	Test Report (TR)	1.0	8/05/2023

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Table of Contents

1. Introduction.....	5
1.1. Purpose.....	5
1.2. Intended Audience.....	5
1.3. Illustrative Example	6
1.4. References	6
1.5. Definitions and Abbreviations	8
2. Frameworks and Languages	9
2.1. Flutter/Dart.....	9
2.2. Null Safety.....	9
2.3. Asynchronous Programming.....	10
2.4. Dart Package Dependencies	10
3. Development Environment	11
3.1. Installing on an Android Emulator on Windows	11
3.1.1. Setting up Android Emulator	11
3.2 Setting Up iOS Simulator for Apple Devices	12
3.2.1 Install Xcode Command Line Tools directly	12
3.3 Cloning Repositories and Creating Issues.....	16
4. Development Process.....	17
5. Code Structure	17
5.1. backend_services Folder Structure.....	17
5.2. References to backend_services Package	20
5.3. backend_test_utility Folder Structure	20
5.4. External Flutter Package Libraries	21
5.4.1. Development Only Dependencies.....	22
6. User Interface.....	23
6.1. Test Utility Main Screen	23
6.2. Test Screens.....	23
7. Data and Backend	24
7.1. RecordingSelectionActivator	24
7.2. .env file.....	26

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

7.3.	Local Storage Strategy	27
7.4.	BESie.....	27
7.5.	S3.....	28
7.6.	AWS Transcribe	28
7.7.	ChatGPT.....	29
8.	Testing.....	29
8.1.	Mockito Dependency	29
8.2.	Browser Extension Test	29
8.3.	Integration Tests	30
8.4.	Testing Focus	30
9.	Known Bugs and Issues	31
10.	Licensing Information.....	31

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

1. Introduction

1.1. Purpose

The Programmer's Guide documentation is intended to explain the technical thinking behind developing the Short-Term Memory System (STeMS) backend app. Developers can use it as a reference and guidance on implementing different features of the building application. Besides Programmer's guide, a suite of documents includes the Project Plan, Software Requirements Specification, Technical Design Document, and Deployment and Operations Guide. These documents can be used in association with this document. In the future, it will provide thorough information about the building application to assist in implementing any new desired feature.

1.2. Intended Audience

The Programmer's Guide documentation is intended for the technical project stakeholders and future programmers who will continue maintaining or developing STeMS applications. The technical project stakeholders are included in the following table:

Table 1, The technical project stakeholders;

Name	Role
Dr. Mir Assadullah	Instructor
Roy Gordon Robert Wilson	Project Mentors
Babers, David	Project Manager
Nagy, Jonathan	Team Lead
Babers, David Hicks, Collin Iko, Benny Mccrillis, Scott	Developers

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Thomare, Amol	
Ben Lakbir, Mohamed De Souza, Robson Huynh, Johnny Mekonnen, Kidanu Nagy, Jonathan	QA/Testers

1.3. Illustrative Example

Given the intended population of users for the application, one scenario that will greatly benefit users is if the user has an important appointment that cannot be missed. With short-term memory loss, the user may remember the appointment for the duration of the current day and then forget that there is a pending appointment the next day. In that scenario, the ConvoBuddy application will be able to record the conversation with all the pertinent information and subsequently create a reminder on the local device. In that manner, the user can always be aware of any appointments and be alerted in a timely manner.

1.4. References

Amazon. (n.d.a). Amazon Transcribe pricing. Retrieved July 15, 2023, from <https://aws.amazon.com/transcribe/pricing/>

Amazon. (n.d.b). Amazon Linux 2023 FAQs. Retrieved July 18, 2023, from <https://aws.amazon.com/linux/amazon-linux-2023/faqs/>

Amazon. (n.d.c). Run command on your Linux instance at launch. Retrieved July 15, 2023, from <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>

Amazon re:Post. (n.d.). How can I utilize user data to automatically run a script with every restart of my Amazon EC2 Linux instance? Retrieved July 17, 2023, from <https://repost.aws/knowledge-center/execute-user-data-ec2>

Chhina, R. (Oct 5, 2021). Creating iOS Simulators for End-to-End Automation Tests. Retrieved July 20, 2023, from <https://medium.com/tauk-blog/creating-ios-simulators-for-end-to-end-automation-tests-6972033ccf1b>

Dart.dev. (n.d.a). Creating packages. Retrieved July 8, 2023, from <https://dart.dev/guides/libraries/create-packages>

Dart.dev. (n.d.b). Sound null safety. Retrieved July 16, 2023, from <https://dart.dev/null-safety>

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Dart.dev. (n.d.c). Understanding null safety. Retrieved July 13, 2023, from <https://dart.dev/null-safety/understanding-null-safety>

Dart.dev. (n.d.d). Asynchronous programming: futures, async, await. Retrieved July 10, 2023, from <https://dart.dev/codelabs/async-await>

Dart.dev. (n.d.e). Package dependencies. Retrieved July 5, 2023, from <https://dart.dev/tools/pub/dependencies>

Flutter. (2023). Racing forward at I/O 2023 with Flutter and Dart. Retrieved July 5, 2023, from <https://docs.flutter.dev/resources/faq>

Flutter. (n.d.a). Windows Install. Retrieved July 9, 2023, from <https://docs.flutter.dev/get-started/install/windows>

Flutter. (n.d.b). Using packages. Retrieved July 9, 2023, from <https://docs.flutter.dev/packages-and-plugins/using-packages>

Flutter. (n.d.c). An introduction to integration testing. Retrieved July 9, 2023, from <https://docs.flutter.dev/cookbook/testing/integration/introduction>

GitHub. (n.d.a). Create an issue. Retrieved July 11, 2023, from <https://docs.github.com/en/issues/tracking-your-work-with-issues/creating-an-issue>

GitHub. (n.d.b). Cloning a Repository. Retrieved July 11, 2023, from <https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository>

Google. (July 18, 2023). Integration testing with Flutter. Retrieved July 14, 2023, from <https://firebase.google.com/docs/test-lab/flutter/integration-testing-with-flutter>

Kohn, R. (Mar 21, 2023). Mastering token limits and memory in ChatGPT and other large language models. Retrieved July 15, 2023, from <https://medium.com/@russtkohn/mastering-ai-token-limits-and-memory-ce920630349a>

Mac. (n.d.). Install Xcode Command Line Tools. Retrieved July 19, 2023, from <https://mac.install.guide/commandlinetools/4.html>

Manferdini, M. (August 4, 2022). The Best and Fastest Ways to Install Xcode on your Mac. Retrieved July 20, 2023, from <https://matteomanferdini.com/install-xcode/>

Najbar, M. (27 Apr. 2020) GitFlow with GitHub. Retrieved July 8, 2023, from <https://medium.com/android-news/gitflow-with-github-c675aa4f606a>

OpenAI. (July 2023). What are tokens and how to count them? Retrieved July 15, 2023, from <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Supervisord.org. (n.d.). Introduction. Retrieved July 20, 2023, from <http://supervisord.org/introduction.html>

1.5. Definitions and Abbreviations

This section covers all abbreviations and acronyms used throughout this Programmer's guide document.

Table 2. Abbreviations and acronyms used throughout the programmer's guide

Abbreviation/Acronym	Definition
AI	Artificial Intelligence
API	Application Programming Interface
BESie	Browser Extension Service
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
I/O	Input and Output (computer hardware)
QA	Quality Assurance
SDK	Software Development Kit
STeMS	Short-Term Memory System
STT	Speech-to-Text
UID	Unique Identifier
VM	Virtual Machine

Table 3. Terms used throughout the programmer's guide

Term	Definition
App Instance Code	A code generated by the STeMS backend system to identify ConvoBuddy application instances and used with the Browser Extension.
ChatGPT	A language model developed by OpenAI that uses deep learning techniques to generate human-like responses in natural language conversations.
ChatGPT API	An interface provided by OpenAI for developers to integrate ChatGPT into their applications
Diarization	The application of an algorithm to separate individual speakers in an audio stream, so that a speaker can be identified with each utterance in a textual transcript.
Local Storage	Storage provided on a user's device, sandboxed on mobile operating systems for security.
Speech-to-Text (STT)	Recognition and translation of spoken language into text. Also known as ASR or computer speech recognition.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

WebSocket	A technology used to provide two-way communication between a web client and a server, allowing event driven responses without having to poll for a reply.
-----------	---

2. Frameworks and Languages

This section lists the frameworks, languages and language features used in the STeMS backend services.

2.1. Flutter/Dart

Flutter is an open-source portable UI toolkit for crafting attractive, natively compiled applications. It allows the unification of app developers into a single mobile, web, and desktop app team, building branded apps for multiple platforms out of a single codebase. Flutter's framework is designed to be layered and customizable. Based on the need, developers can choose to use only parts of the framework or even replace the upper layers of the framework entirely. It is combined with Dart applications to build fast, natively compiled applications. Concerning implementing multiplatform applications, Flutter prevents the hassle of writing separate codebases for each platform, such as Android and iOS. Therefore, it saves development effort by applying a single codebase (Flutter, 2023).

Dart is a free and open-source programming language. It is designed for multiplatform application development. Dart is adopted for the framework and widgets based on four primary dimensions for evaluation and considers the needs of framework authors, developers, and end users. It is improved for use in Flutter, like optimizing the VM for latency, which is more significant for Flutter's workload. Dart is a better fit to work with Flutter for the following primary criteria; developer productivity, object orientation, predictability, high performance, and fast allocation (Flutter, 2023).

Flutter SDK

According to Flutter (2023), Flutter SDK includes:

- Heavily optimized, mobile-first 2D rendering engine with excellent support for text
- Modern react-style framework
- Rich set of widgets implementing Material Design and iOS-style
- APIs for unit and integration tests
- Interop and plugin APIs to connect to the system and 3rd-party SDKs
- Headless test runner for running tests on Windows, Linux, and Mac
- Flutter DevTools or Dart DevTools for testing, debugging, and profiling the app
- Command-line tools for creating, building, testing, and compiling applications

2.2. Null Safety

Dart provides type safety with a static type system and null safety by making reference values non-nullable by default (Dart.dev, n.d.b). This differs from traditional reference value-based

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

languages, where a statically typed variable could be null or point to an instance of the specified type. This removes an entire class of null reference errors from even being possible when adequately utilized.

A variable can be made nullable by adding a question mark (?) at the end of the type annotation. Traditional runtime null reference errors are turned into “edit-time” analysis errors (dart.dev, n.d.c), thanks to the dart code analysis integrated into modern IDEs. “Soundness” of null safety is a term used to indicate that if “an expression has a static type that does not permit null, then no possible execution of that expression can ever evaluate to null” (dart.dev, n.d.c).

2.3. Asynchronous Programming

The Dart language provides a simple programming model for interacting with asynchronous method calls, using the keywords Future, async, and await. Dart code written using asynchronous keywords looks and, for the most part, acts like regular synchronous code, but with the performance benefits of non-block I/O and utilization of multiple processing cores. This means Dart asynchronous code is both readable and performant.

Asynchronous functions execute synchronously up until the first await keyword, and then execution diverges. Code reliant on the awaited value is blocked until the future resolves, while code not dependent upon the awaited result is executed immediately. The await keyword allows the use of try/catch blocks as if the methods were running synchronously. This simplifies error handling and provides code that looks synchronous but executes more efficiently.

2.4. Dart Package Dependencies

Various external libraries were used to help develop the STeMS backend services. Eternal Flutter packages allow developers to utilize the contribution of other developers in their Flutter applications (Flutter, n.d.b). Pub.dev provides a web interface to search for packages, which can be searched for and filtered by Android, iOS, and Web platforms.

Packages can be added to a flutter/dart project by either adding the package name under the “dependencies:” section or by entering the command flutter pub add <package_name> at the project root. Package version can be indicated in the pubspec.yaml with either the version number of a version range in caret (^) syntax or using <, >, <=, or >=. A caret in front of a version number means all versions backwards compatible with the indicated version number, or example, ^1.2.3 is equivalent to '>=1.2.3 <2.0.0' (Dart.dev, n.d.e).

Suppose no version number is specified for a package dependency in the pubspec.yaml. In that case, the package manager will use the highest available version that satisfies all the package version requirements in the other dependencies in the pubspec.yaml file. In the following example, the first package dependency indicates the latest version greater than or equal to 5.4.0, but less than the next major version, 6.0.0, which could have breaking changes. The second

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

example specifies exactly version 5.4.3. If another dependency relies on a different version of image_picker, the package manager will fail to update dependencies on flutter pub get.

Figure 1. Dependencies Versions Examples

```
dependencies:
  url_launcher: ^5.4.0    # Good, any version >= 5.4.0 but < 6.0.0
  image_picker: '5.4.3'  # Not so good, only version 5.4.3 works.
```

(Flutter, n.d.b)

3. Development Environment

[IDE(s), getting started: cloning repository, setting up android emulator, setting up iOS simulator, link to code repository, link to issues]

This section discusses the developer environment used to write and test the STeMS backend services and setup procedures for that environment. Additionally, aspects of the development process are included, such as work package assignment and issue resolution.

The repository

3.1. Installing on an Android Emulator on Windows

If you already have an emulator setup, you can skip the next section and go to Installing ConvoBuddy on Android Emulator.t.

3.1.1. Setting up Android Emulator

You will need Android Studio installed to create an emulated Android device.

3.1.1.1. Install Android Studio

The first step to setting up an Android emulator is to install Android Studio along with the necessary SDK build tools.

1. Download and install [Android Studio](#).
2. Start Android Studio and go through the ‘Android Studio Setup Wizard’. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.
3. Run flutter doctor to confirm that Flutter has located your installation of Android Studio. If Flutter cannot locate it, run `flutter config --android-studio-dir=<directory>` to set the directory that Android Studio is installed to.

(Flutter, n.d.a)

3.1.1.2. Optionally enable Hardware Acceleration

Developers have the option to enable hardware acceleration to increase performance of the Android Emulator, with the caveat that you may run into hardware incompatibility or graphics

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

issues. You can read me and find instructions at <https://developer.android.com/studio/run/emulator-acceleration#accel-vm>.

3.1.1.3. *Set up an Emulator*

The following instructions from flutter.dev website explain how to set up an Android emulator using Android Studio.

1. Launch **Android Studio**, click the **Device Manager** icon, and select **Create Device** under **Virtual** tab...
 - o In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device....** (The **Android** submenu is only present when inside an Android project.)
 - o If you do not have a project open, you can choose **3-Dot Menu / More Actions > Virtual Device Manager** and select **Create Device...**
2. Choose a device definition and select **Next**.
3. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86_64* image is recommended.
4. Under Emulated Performance, select **Hardware - GLES 2.0** to enable [hardware acceleration](#).
5. Verify the AVD configuration is correct, and select **Finish**.

For details on the above steps, see [Managing AVDs](#).

6. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

(Flutter, n.d.a)

3.2 Setting Up iOS Simulator for Apple Devices

This section covers how to create and set up an iOS Simulator using Xcode for Apple devices.

3.2.1 Install Xcode Command Line Tools directly

Xcode is an integrated development environment (IDE). It's an application with comprehensive functionality that helps developers write code and create apps. Xcode only runs on macOS.

There is no Windows version. However, Visual Studio Code (Vscode) is an alternative (Manferdini, 2022). Xcode Command Line Tools are tools for software developers that run on the command line, in the Terminal application. According to Mac (n.d.), Many developers set up a local development environment with Homebrew. Its installation process offers an option to

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

install Xcode Command Line Tools ([Install Xcode Command Line Tools with Homebrew](#)). Yet, there is a way to install it directly without Homebrew.

Steps to follow:

- Check and update macOS version.
- Check if Xcode Command Line Tools was previously installed.
- Use `xcode-select --install` to install Command Line Tools performing the following instructions.

These instructions are for a terminal running Zsh, the Z shell, on a newer Mac. Older Macs may be running the bash shell and you should upgrade.

Triggering installation of Xcode Command Line Tools

Apple macOS will prompt to install Xcode Command Line Tools using git – a version control system –.

Use 'xcode-select' to install Xcode Command Line Tools

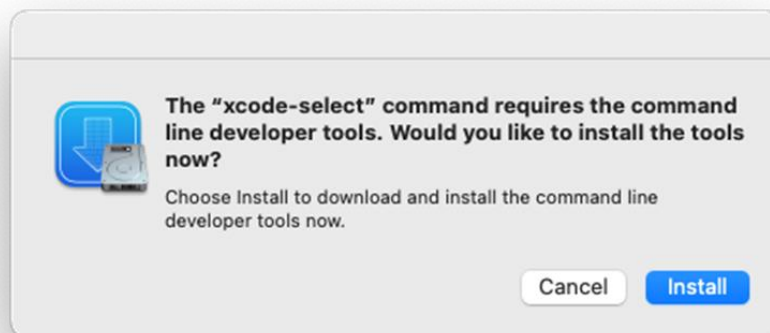
The command `xcode-select` is used to install Xcode Command Line Tools with the command `xcode-select --install`.

The command `xcode-select --install` will open a dialog for installation of the command line tools.

```
$ xcode-select --install
```

A panel that asks to install Xcode Command Line Tools will be displayed.

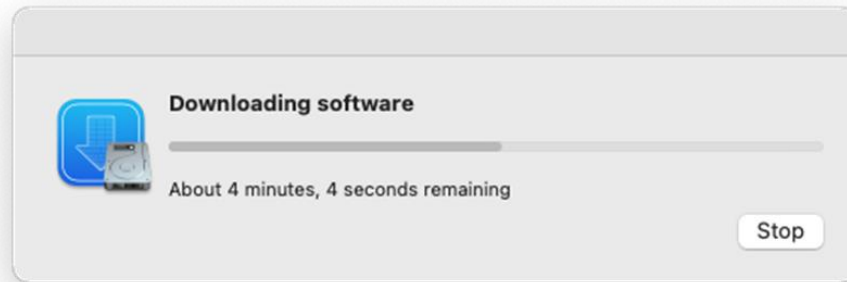
Figure 2. Install Xcode Command Line Tools



Click 'Install' to begin the download and installation process.

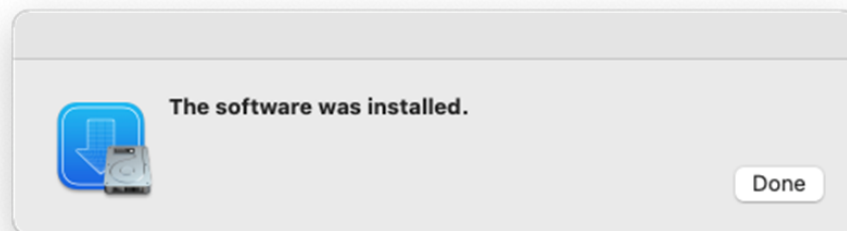
STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 3. Downloading Xcode



Xcode Command Line Tools installation from the prompt panel takes few minutes.

Figure 4. Xcode Installed



A confirmation message when installation is complete will be displayed.

Verify installation of Xcode Command Line Tools

To verify that Xcode Command Line Tools is successfully installed.

```
$ xcode-select -p
/Library/Developer/CommandLineTools
```

To check that you can run git:

```
git --version
git version 2.40.0 (Apple Git-130)
```

3.2.2 Setting Up an iOS Simulator

After installing Xcode in a Mac OS, this section covers how to create and launch standard and custom iOS Simulator.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Creating a standard Simulator

To open a standard Simulator, open Xcode and click on “Xcode” in the menu bar. Then, hover over “Open Developer Tool” and then select the “Simulator” option. An iOS Simulator boots up on the screen.

Figure 5. iOS Simulator



Creating custom Simulator

To create another simulator;

Click on “File” > “New Simulator...” in the menu bar. This will open another window with the *New Simulator* GUI. This GUI allows you to enter a name and then select the Device Type and OS Version for the new Simulator to create.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 66. New Simulator Dialog



New Simulator GUI Wizard will be created. To launch it, click on “File” > “Open Simulator” > OS Version > Name of the simulator you entered. Usually, it takes about a few minutes for the simulator to start upon first creation (Chhina, 2021).

3.3 Cloning Repositories and Creating Issues

This section talks about the repository can be cloned and how issues can be opened for specific repositories

The repository used in this case is GitHub. The GitHub repository can be cloned from a web browser onto a local machine by using the following steps:

- **Go to the main repository page at GitHub.com.**
- **Click the green [Code < >] button in the upper right corner of the page above the code.**
- **Copy the url that is listed there.**
- **Open up a Git Bash terminal on your local machine.**
- **Go to the directory you would like the repository to cloned to.**
- **Enter the following Command: `git clone https://github.com/url/to/your/repository/` and click Enter.**

By following these directions, the repository can be cloned into your local environment. (GitHub (n.d.b))

In addition, issues can be created for specific repositories by following these instructions:

- **Navigate to the main web page of the repository of the issue**
- **Click the issues tab under the name of the repository**
- **Click New issue and click the issue type or open a blank issue depending on the issue in question**

Type out the title and description of the issue and click submit.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

The issue is then created for the specific repository. In addition, a project maintainer can assign the issue to a person, add it to a project board, associate it with a milestone, or apply a label. (GitHub (n.d.b))

4. Development Process

The application will be developed by worked on by multiple developers in different locations. This would require the use of a common repository for developers to push their code up to. The repository chosen for this application is GitHub, which is a shared repository application. The repository uses merging, pull requests, and code reviews to share a joint code base among multiple developers. Merging one branch into another involves taking all the changes made to the derived branch and applying them all to the merged branch in question. Pull requests are when a developer is ready to submit changes that they've made to a branch to the main branch. Code reviews are when other developers look at the changes made by one developer to ensure that it won't break the code and that it makes sense.

Specifically, GitHub is used in a process called GitFlow, which allows developers to have a single development branch in which features can be tested in feature branches that are derived from the development branch. The tested features are merged back into the main development branch, where it is branched off again to test the developed application as a whole by QA. Any errors found by QA are fixed and merged into development before development is merged into the origin/master branch. (Najbar, M. (27 Apr. 2020))

5. Code Structure

The backend_services project is a Dart package that is portable and platform agnostic. System libraries are referenced through common dart abstractions, such as path_provider, which supports mappings for directory locations across the various Flutter platforms, including Android, iOS, Linux, macOS, and Windows operating systems. The portable nature of backend_services means that it can be used to build applications on any of the platforms mentioned, just like many Flutter packages available on the Flutter package site pub.dev.

To test the application programming interfaces exposed by the backend_services package, a backend_test_utility was created. It includes flutter platform folders for Android and iOS to enable testing backend services on those platforms. The web platform folder is included as well, as it provides a quicker build time when working with the backend_test_utility user interface, though file IO is not supported in web builds by all the packages used by backend_services.

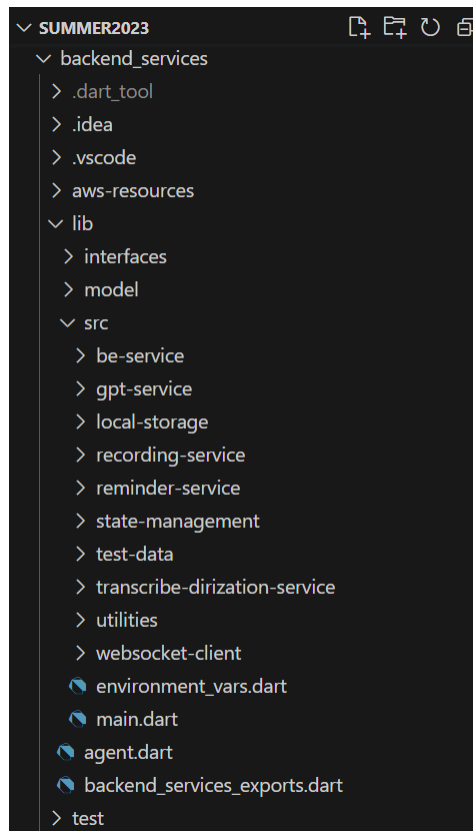
5.1. backend_services Folder Structure

The folder structure of backend_services is based on the dart template for packages and does not include various platform folders commonly associated with Flutter projects. This keeps the

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

backend_services project simple and puts focus on the code. The project folder structure is made up of project source code in the lib folder, unit test files in the test folder, and the AWS-resources folder for AWS deployed scripts and functions used for AWS Transcribe and diarization.

Figure 77. backend_services Package Folders



The bulk of backend_services functionality is exposed by agent.dart, which contains all the methods for the various services. Functionality, including—but not limited to—restaurant order summary, STML reminder generation, and browser extension form filling, is called through an instance of Agent given a conversation UID and some additional information outlined in the method signatures.

The interface folder contains two abstract classes used primarily to help test conversation selection and storage of data to JSON files, JSONStorage and RecordingSelectionActivator. These abstract classes provide interfaces that are implemented in real classes but also fakes used in backend_service unit tests and backend_test_utility integration tests.

Interface: abstract class JSONStorage

- Purpose: Provides an interface for writing data to a conceptual set of files in a folder, given the file name and a data class instance.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

- Real implementation: class LocalStorage
- Fake implementation: class TestStorage
- Interface: abstract class RecordingSelectionActivator
 - Purpose: Provides an interface for a callback method that allows the front end to activate a conversation selection screen when required by the backend.
 - Real implementation: class DummyConversationSelectionActivator (in talker-mobile-app, full implementation with selection screen pending)
 - Fake implementations: classes TestRecordingSelectionActivator and TestSelectionAndExtractionActivator

The model folder contains all the data models used for backend_services storage and communication, such as browser extension service request and response classes BERequest and BEResponse, the Conversation class, Reminder class, and User class.

The lib/src folder is intended by dart language convention to contain code not to be directly exposed to external packages and applications (dart.dev, n.d.a). Conversely, dart files in the lib folder root are intended to be exposed, and specifically Agent, as mentioned, acts as the primary entry point for all backend_services functionality. The file backend_services_exports.dart selectively exposes classes under the lib/src folder and allows external clients of backend_services to include all backend_services exposed classes with one import, `import 'package: backend_services/backend_services_export.dart'`.

Folders and files under lib/src represent the implementation of backend_services including:

- **be-service** – Methods for supporting the Browser Extension API, including storing of request and parsing data from websocket frames.
- **gpt-service** – Contains GptCalls, where prompt generation and calls to OpenAI's chat completion service are implemented.
- **local-storage** – Implementation local storage service
- **recording-service** – placeholder for recording service
- **reminder-service** – placeholder for reminder service
- **state-management** – Contains the conversation_provider.dart file relocated from Team A's talker-mobile-app flutter application. This class provides an interface for both the front and backend to access recording files and recording metadata.
- **test-data** – Contains conversation test data for use with unit tests, integration tests, and the backend_test_utility application.
- **transcribe-diarization-service** – Contains methods for uploading audio files to AWS for speech-to-text transformation with AWS Transcribe
- **utilities** – contains a json_utils.dart for generating “pretty” indented JavaScript Object Notation (JSON) formatted data.
- **websocket-client** – Contains the websocket_client.dart which connects to BESie using WebSocket protocol. The websocket_listener.dart provides an object for configuring a callback for messages on a particular WebSocket topic.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

- **environment_vars.dart** – At the root of the lib/src folder, this file provides a common dart interface to all the environment variables configured through the .env file in the project root. Variables can be coded to allow overriding by the github Backend CI workflow.
- **main.dart** – A test application entry point, currently just housing an instantiation call to the RecordingService(), which is also a placeholder.

5.2. References to backend_services Package

Currently there are two flutter applications that reference the backend_services package: backend_test_utility and talker-mobile-app. Both include the relative path of the backend_services package source in their respective pubspec.yaml files under dependencies.

Figure 8. pubspec.yaml backend_services Reference Example

```
dependencies:
  flutter:
    sdk: flutter

# backend services package
backend_services:
  path: ../backend_services
```

5.3. backend_test_utility Folder Structure

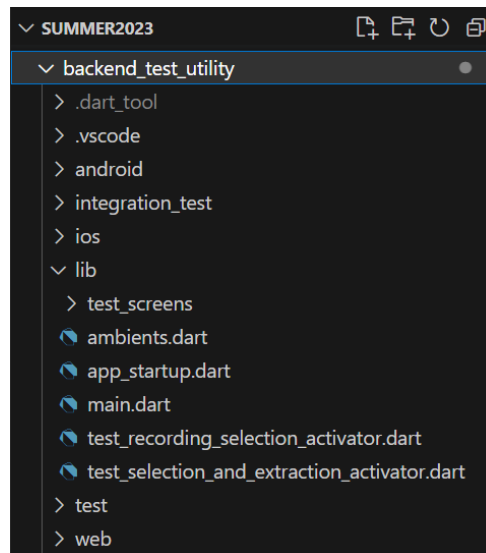
The backend_test_utility project folder structure is based on the Flutter app template with platform build folders for Android, iOS, and the web. Android and iOS are the target platforms, but the web was included as well, as the web platform build compiles more quickly and is handy in developing UI quickly. As this project focuses on testing, the web provides a quicker development experience when platform-specific concerns like I/O are not necessary. Like the backend_services dart package, the backend_test_utility contains:

- Implementation code in the lib folder.
- Unit tests are in the test folder.
- Additionally, integration tests are in the integration_test folder.

The root of the backend_test_utility project contains a .env.readme with instructions on setting up a .env file for use on the developer's machine and a .env.release used as the .env for release builds during the GitHub Backend CI build workflow.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 9. backend_services Package Folders



Folders and files under lib include:

- **test_screens** – a set of simple screens for testing individual backend_services functionality.
- **ambients.dart** – Global variables, including the getIt instance and WidgetKeys to assist in automated testing of test_screens
- **app_start.dart** – Registers a singleton Agent and loads the .env file using package flutter_dotenv.
- **main.dart** – backend_test_utility app entry point, including the main screen UI.
- **test_*_activator.dart** – Two classes that provide a callback that logs and/or simulates response from the front-end on a request to select a conversation from the backend.
- **test** – unit tests for backend_test_utility. Essentially empty, given the integration_test focus of backend_test_utility.

5.4. External Flutter Package Libraries

The following packages were used in the development of the STeMS backend services projects.

Table 4. Flutter Packages Used in backend_services

Package Name	Version	Description
dart_openai	^3.0.0	Provides access to OpenAI's ChatGPT and Whisper APIs. Fairly popular and recently updated.
flutter_dotenv	^5.1.0	Load configuration at runtime from a .env file.
foundation	^0.0.5	Used for state management for the ConversationProvider class.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Package Name	Version	Description
logger	^1.4.0	Used to log various debug and error information during application execution.
path	^1.8.3	Cross-platform path manipulation library.
path_provider	^2.0.15	Provides commonly used locations on the filesystem for most platforms.
stomp_dart_client	^0.4.4	A dart client for connecting to a remote STOMP server.
uuid	^3.0.6	Simple fast generation of RFC4111 UUIDs.
mockito	^5.4.2	A mock framework inspired by Mockito with APIs for Fakes, Mocks, behavior verification, and stubbing.
build_runner	^2.4.6	A build system for Dart code generation and modular compilation. Used by Mockito to generate mock classes.
http	^0.13.6	A composable, multi-platform, Future-based API for HTTP requests. Latest version 1.1.0 not compatible with openai_dart, so 0.13.6 used instead.
collection	^1.17.1	Collections and utilities functions and classes related to collections.

Table 5. Flutter Packages Used in backend_test_utility

Package Name	Version	Description
cupertino_icons	^1.0.2	Default icons asset for Cupertino widgets based on Apple styled icons.
logger	^1.4.0	Used to log various debug and error information during application execution.
get_it	^7.6.0	Simple direct Service Locator that allows to decouple the interface from a concrete implementation and to access the concrete implementation from everywhere in your App
flutter_dotenv	^5.1.0	Load configuration at runtime from a .env file.
path_provider	^2.0.15	Provides commonly used locations on the filesystem for most platforms.
collection	^1.17.1	Collections and utilities functions and classes related to collections.

5.4.1. Development Only Dependencies

These dependencies are only used during development and are left out of release builds.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Table 6. Flutter Dev Packages Used

Package Name	Version	Description
flutter_test	SDK version (3.10.5)	Provides access to OpenAI's ChatGPT and Whisper APIs. Fairly popular and recently updated.
integration_test	SDK version (3.10.5)	Runs tests that use the flutter_test API as integration tests.
lints	^2.0.0	Static analyzer for detecting problems with code.
test	^1.21.0	The dart package for writing unit tests.
flutter_lints	^2.0.0	Recommended lints for Flutter apps, packages, and plugins to encourage good coding practices.

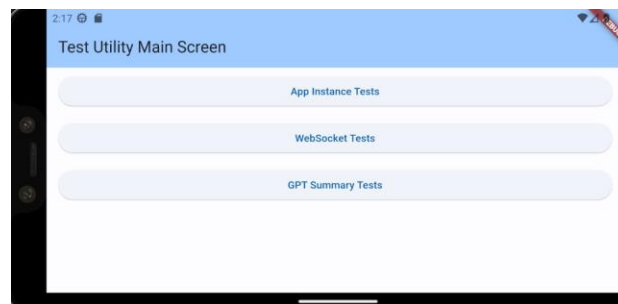
6. User Interface

The ConvoBuddy user interface is the responsibility of Team A, and thus out of scope of this document. The backend_test_utility does have a simple user interface and that will be briefly discussed here.

6.1. Test Utility Main Screen

The backend_test_utility main screen contains a button for each test screen.

Figure 10. backend_test_utility Main Screen

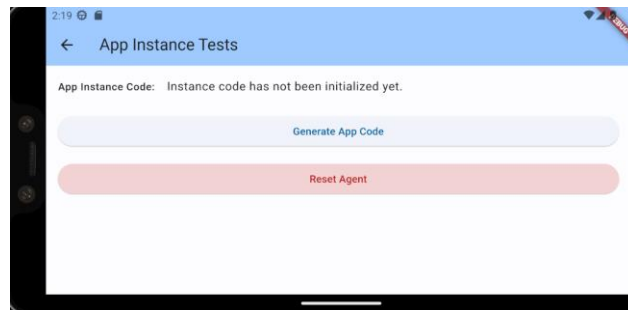


6.2. Test Screens

Test screens exercise functionality of a single backend_services method or set of related methods. The example below tests the app instance code generation.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 11. App Instance Test Screen



7. Data and Backend

[local storage strategy, BESie, AWS Transcribe, S3, ChatGPT, .env for api keys]

7.1. RecordingSelectionActivator

The RecordingSelectionActivator interface specifies a `Future<void> getSelectorCallback()` method.

The front-end application can provide a callback to the backend to trigger a conversation selection screen for the form filler browser extension. On selecting a conversation on this screen, the front-end can call `Agent.extractFormValues()` with the selected conversation UID to kick off completing the form fill request.

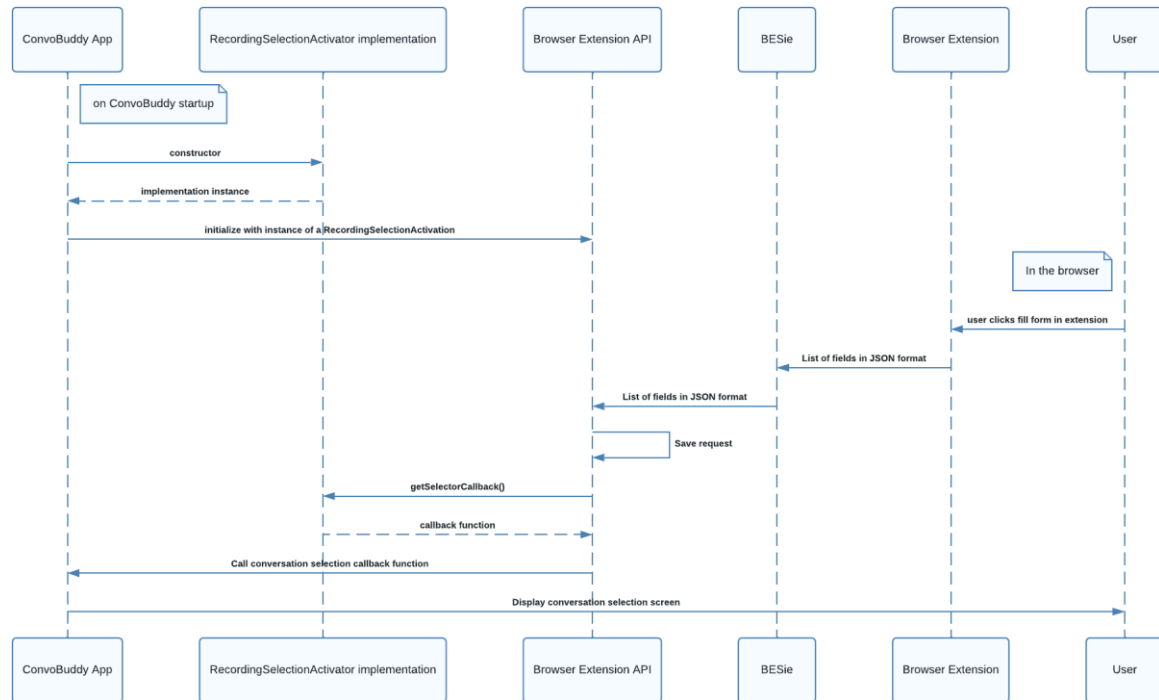
A RecordingSelectionActivator implementing object is required to initialize the Agent as a parameter to `Agent.initialize()`.

Expected flow is:

1. The browser extension API receives a request over BESie on the fill-form topic.
2. The API verifies the app instance code, stores the list of requested fields to fill for use later, and calls the previously provided RecordingSelectionActivator instance.
3. The activator is responsible for displaying the UI for the user to select a recording, and that UI is responsible for calling `Agent.extractFormValues()` with the selected recording guid.
4. The browser extension API will query ChatGPT and send the results back to BESie on the filled-form topic.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 12. RecordingSelectionActivator Sequence Diagram



STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 13. Example RecordingSelectionActivator Implementation

```
class ExampleConversationSelectionActivator
  implements RecordingSelectionActivator {
  ExampleConversationSelectionActivator(this.rootContext);

  BuildContext rootContext;

  @override
  RecordingSelectionActivatorCallback getSelectorCallback() {
    return () async {
      // Ensure context is still mounted before displaying dialog
      if (!rootContext.mounted) return;
      Navigator.pushNamed(rootContext, '/conversationSelectionScreen');
    };
  }
}
```

Figure 14. Call to RecordingSelectionActivator Implementation in agent.dart

```
Future<void> receiveFormValuesRequest(BERequest request) async {
  final appCode = await _beService.loadAppInstanceCode();
  if (request.pin != appCode) {
    _logger.i(
      "Ignoring browser extension request with pin of '${request.pin}'.");
    return;
  }
  _beService.storeRequest(request);

  // trigger recording selection UI
  var callback = _recordingSelectionActivator!.getSelectorCallback();
  return callback();
}
```

7.2. .env file

The flutter_dotenv package is used to import configuration entries into the application from a .env file deployed with the application. The flutter_dotenv package is included under dependencies in the pubspec.yml, and the .env is included under the asset section. When API secrets are to be included in the .env file, the .env file is generally added to the .gitignore file a project, and no .env is checked in. This is the case for backend_test_utility, and developer instructions to set up a .env file are included in a .env.readme.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 15. Setup of .env in the pubspec.yaml

```
dependencies:
  flutter:
    sdk: flutter

  ...

  flutter_dotenv: ^5.1.0

  ...

flutter:
  assets:
    - .env
```

A .env.release file is also included in source control, intended to be copied to .env file during the github Backend CI build workflow. It includes various configuration items for the release build and an empty OPENAI_API_KEY, which is overridden by the github workflow by setting --dart-define=OPENAI_API_KEY="{{secrets.OPENAI_API_KEY}}" in the apk build action.

Figure 16. .env.release File Contents

```
backend_test_utility > $ .env.release
1  # This .env.release file is used by the github build workflow as a base set of environmental
2  # variables that can be added to or updated with the --dart-define flutter argument.
3
4  RECORDING_S3_BUCKET = 'https://testrecordingsswenv2.s3.amazonaws.com'
5  TRANSCRIPTION_S3_BUCKET='https://transcriptsswenv1.s3.amazonaws.com'
6
7  # BESie on the internet
8  WS_URL = 'http://44.202.25.184:8080/ws'
9  WS_CONNECTION_TIMEOUT_MS = '10000'
10 FORM_FILL_REQUEST_TOPIC = '/topic/form-model'
11 FORM_FILL_RESPONSE_TOPIC = '/app/filled-form'
12
13 # Do not populate, this is set in the github workflow
14 OPENAI_API_KEY =
15 |
```

7.3. Local Storage Strategy

For simplicity and testability, local file storage was used for storing various entities in the application. The intention was to avoid complexity and setup costs of a mobile database or deployment of remote database or services for storage of these application entities. JSON was the data storage format of choice, due to its readability, ubiquitous tooling, and better storage profile than other technologies such as XML.

7.4. BESie

BESie is the Browser Extension Service and provides a link between the Flutter app on a mobile device and the browser extension running on another device. As neither a web browser or a mobile device is a particularly good platform for hosting a server, an intermediary was designed to broker connections between the two. Allowing two way communication between both clients, WebSockets were used. A WebSocket server provides long-lived connections that

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

will enable event-driven messaging from server to client and typical client-to-server requests. The ConvoBuddy mobile app connects to BESie on startup and begins listening for requests from the Browser Extension. Likewise, when the browser extension is started, it begins listening for responses from the ConvoBuddy app. Communication is not direct, but WebSocket servers provide a fast and always connected pathway between participating clients.

The utilities Amazon cloud-init, system, and supervisors were considered for running BESie on startup, reboot, and if the BESie process should crash. Amazon cloud-init scripts “run only during the first boot cycle when an EC2 instance is launched,” though a workaround exists (Amazon re:Post, n.d.). Supervisord was included in Amazon-Linux-Extras, but the package is not currently supported by the latest Amazon Linux image, AL2023 (Amazon, n.d.b). Systemd is both easy to use and integrates well with journald for viewing and managing logs. However, it is not as easy to use as Amazon’s cloud-init with a web interface (Amazon, n.d.c), or Supervisord, focusing on enabling non-privileged users to create and manage services (Supervisord.org, n.d.).

7.5. S3

AWS was selected as the primary source of external services needed to properly run the ConvoBuddy application. The reason for choosing AWS was the ease of use for the development team members that already had an account with free-tier services available and had prior experience with other services provided by AWS.

All the recordings and their respective transcriptions are all saved to a separate S3 bucket. The location for these buckets is hard coded into the .env files to ensure a proper communication link can be maintained. This alleviates any burden in the local storage device and the free tier provides enough capacity where the users will not have any issues in saving and then recalling any specific conversation or transcription.

7.6. AWS Transcribe

AWS Transcribe is a speech-to-text service that will allow the saved conversation recordings and create a text file that can be used as a reference for forgotten details or for further analysis, as is the case for this application. The free tier offering allows for up to 60 minutes of audio files to be transcribed, which is enough for the purposes of testing.

Since the decision was made to utilize AWS for external services, AWS Transcribe was the perfect choice for converting the audio recordings into a text format to be analyzed by ChatGPT. Again, the team leveraged the free tier to run initial tests with audio recordings and ultimately chose to proceed with the service. The quality of the transcriptions was high enough for an external viewer to understand the context and content of the conversations. Other services, such as Google’s speech-to-text offering, were considered, but were not chosen due to the simplicity of hosting data and service in a central location, i.e. AWS.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

7.7. ChatGPT

ChatGPT is a groundbreaking technology that has allowed near universal access to large language model (LLM) driven artificial intelligence (AI). The OpenAI organization makes ChatGPT free to use through their website as a freemium business model, with a higher grade version of the GPT LLM model available through a subscription fee.

More relevant to this project, OpenAI has ample services and APIs available to interface with ChatGPT and related technologies such as Whisper API, a speak-to-text service. Access to OpenAI's services is charge per token of usage, but costs are generally low, especially for the older 3.5 GPT model STeMS backend services is utilizing, and \$5 of free credits is available upon registration. The GPT 3.5-turbo model was chosen due to its speed and relatively low cost per token.

ChatGPT is utilized to generate natural language responses to queries about transcripts taken from audio recorded by the ConvoBuddy application. The key use cases ChatGPT has helped us implement are extracting reminders with timestamps from recorded audio, summarizing the food order of a group of people in a natural setting with requests in arbitrary order and interspersed, and finally extraction of web form field values from information contained in pre-recorded audio. ChatGPT has proven proficient and generally accurate in pulling information from a transcript if it is available and the prompt is properly written.

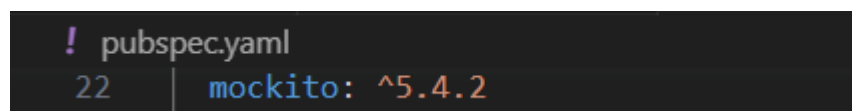
8. Testing

This section discusses the testing approach and tools used to conduct testing.

8.1. Mockito Dependency

In the `backend_services` package, the file `pubspec.yaml` contains all dependencies used in this package. In this file, `mockito` is defined as a dependency under the `dependencies` section. Each test file using Mockito will import this dependency.

Figure 17. Mockito Version



```
! pubspec.yaml
22 | mockito: ^5.4.2
```

8.2. Browser Extension Test

In the test folder of the `backend_services` package, test files are implemented to test certain functionalities. In the context of Mockito, mocks are used to simulate the behavior of real objects through Mockito's API to create mock objects. For example, with testing the browser extension API, the parameter `RecordingSelectionActivator` is passed into a mock for unit tests. The syntax follows an annotation format such as `@GenerateNiceMocks([MockSpec<RecordingSelectionActivator>()])`. In the context of the browser extension, the `RecordingSelectionActivator` is a callback on a selection from the front end for a conversation.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Three tests are performed on the browser extension functionality:

- Receiving and validating requests from BESie
- Querying ChatGPT
- Sending results back to BESie

The mock object created in the test file is used to verify the list of functionalities. The mock object is passed into the verify() method to validate if the RecordingSelectionActivator mock object's method was called by returning true for the callback object.

8.3. Integration Tests

The Flutter SDK provides integration testing as a built-in feature. Flutter unit tests already provide testing of widgets and individual UI components, but integration testing allows testing of the entire app on an emulator or even target hardware (Flutter, n.d.c). Using the same widget UI testing mechanisms as unit tests, complex testing usually requiring a third-party tool like selenium can be written and executed all within the IDE.

Integration tests exist under the backend_test_utility project in the integration_test folder. This is configuration by convention, as any files under the integration_test folder ending with test.dart are considered integration tests by the SDK and when run will run on connected real or virtual hardware. Google even offers Flutter integration testing in the cloud with Firebase Test Lab (Google, 2023).

8.4. Testing Focus

In consideration of options for how to proceed with the Testing phase of the project, there possibilities were considers. Integration_tests using Flutter SDK, a Testing Utility application, and directly testing ConvoBuddy App and Browser Extension. In order to cover the most amount of application surface, in conjunction with Team A, direct testing was determined to be the focus. The backend_test_utility was introduced as a development aid and has helped validate backend_services are working and maintain better organized and testable code.

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

Figure 18. Testing Options

	Integration_test backend_services	Testing Utility	ConvoBuddy App / Browser Extension
Runs on Native device	✓	✓	✓
Can be automated	✓	✓	!
Help Team A cover gaps	✗	✗	✓
Allow for manual modifying tests	✗	!	✓
No coding required	✗	✗	✓
Dependent on Team A	✓	✓	✗
Can run a build without the Dev Environment	✗	✓	✓
Requires a working dev environment	✗	✓	✓

9. Known Bugs and Issues

- Chat GPT 3.5 Model is limited to 4096 tokens (Kohn, 2021). A token representing approximately 3/4 of word and this limit is shared between prompt and completion (OpenAI, 2023). If a transcript is too long, it may exhaust all the tokens as part of the prompt before ChatGPT can generate a complete response. One workaround would be to summary transcripts and use the summary for prompts where tokens would otherwise be exhausted.
- AWS Transcribe Free Tier is limited to 60 hours of transcription a month (Amazon, n.d.a)
- Transcription size may be limited by AWS Lambda transmission limits. Initial testing resulted in a 1009 error when sending transcripts of audio over 1 minute length. There may be configuration or workaround to address issue forthcoming.

10. Licensing Information

AWS Service Terms

<https://aws.amazon.com/service-terms/>

Below are Universal Service Terms (Applicable to All AWS Services) and govern use of the Services. For purposes of these Service Terms, “Your Content” includes any “Company Content” and any “Customer Content,” and “AWS Content” includes “Amazon Properties.”

- Any software (including related documentation) obtained from AWS or third-party licensors in connection with the Services may not be transferred outside the Services without specific authorization to do so.
- Current technical documentation applicable to the Services (including applicable user, admin, and developer guides) posted on the AWS Site at

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

<https://docs.aws.amazon.com/index.html> (and any successor or related locations designated by AWS) must be complied with.

- Information or other materials related to Your Content (including copies of any client-side applications) must be provided, as reasonably requested by AWS to verify compliance with the Agreement. Reasonable cooperation is a must to identify the source of any problem with the Services that AWS reasonably believes may be attributable to Your Content or any end user materials that you control.
- In connection with use of the Services, end users are responsible for maintaining licenses and adhering to the license terms of any software they may run. If AWS reasonably believes any Content violates the law, infringes or misappropriates the rights of any third party, or otherwise violates a material term of the Agreement (including the Service Terms, or the Acceptable Use Policy) (“Prohibited Content”), we will notify you of the Prohibited Content and may request that such content be removed from the Services or access to it be disabled. If you do not remove or disable access to the Prohibited Content within 2 business days of our notice, AWS may remove or disable access to the Prohibited Content or suspend the Services to the extent AWS is not able to remove or disable access to the Prohibited Content. Notwithstanding the foregoing, AWS may remove or disable access to any Prohibited Content without prior notice in connection with illegal content, where the content may disrupt or threaten the Services or in accordance with applicable law or any judicial, regulatory or other governmental order or request. If AWS removes Your Content without prior notice, they will provide prompt notice to you unless prohibited by law. AWS terminates the accounts of repeat infringers in appropriate circumstances.
- You must ensure that all information you provide to AWS via the AWS Site (e.g., information provided in connection with your registration for the Services, requests for increased usage limits) is accurate, complete, and not misleading.
- From time to time, AWS may apply upgrades, patches, bug fixes, or other maintenance to the Services and AWS Content (“Maintenance”). AWS agrees to use reasonable efforts to provide you with prior notice of any scheduled Maintenance (except for emergency Maintenance), and you agree to use reasonable efforts to comply with any Maintenance requirements that AWS may notify you about.
- If your Agreement does not include a provision on AWS Confidential Information, and you and AWS do not have an effective non-disclosure agreement in place, then you agree that you will not disclose AWS Confidential Information (as defined in the AWS Customer Agreement), except as required by law.
- You may perform benchmarks or comparative tests or evaluations (each, a “Benchmark”) of the Services. If you perform or disclose, or direct or permit any third party to perform or disclose, any Benchmark of any of the Services, you (i) will include in any disclosure, and will disclose to AWS, all information necessary to replicate such Benchmark, and (ii) agree that AWS may perform and disclose the results of Benchmarks of your products or

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

services, irrespective of any restrictions on Benchmarks in the terms governing your products or services.

- Only the applicable AWS Contracting Party (as defined in the AWS Customer Agreement) will have obligations with respect to each AWS account, and no other AWS Contracting Party has any obligation with respect to such account. The AWS Contracting Party for an account may change as described in the Agreement. Invoices for each account will reflect the AWS Contracting Party that is responsible for that account during the applicable billing period.
- If, as of the time of a change of the AWS Contracting Party responsible for your account, you have made an up-front payment for any Services under such account, then the AWS Contracting Party you paid such up-front payment to may remain the AWS Contracting Party for the applicable account only with respect to the Services related to such up-front payment.
- When you use a Service, you may be able to use or be required to use one or more other Services (each, an “Associated Service”), and when you use an Associated Service, you are subject to the terms and fees that apply to that Associated Service.
- If you process the personal data of End Users or other identifiable individuals in your use of a Service, you are responsible for providing legally adequate privacy notices and obtaining necessary consents for the processing of such data. You represent to us that you have provided all necessary privacy notices and obtained all necessary consents. You are responsible for processing such data in accordance with applicable law.
- If you have been charged for a Service for a period when that Service was unavailable (as defined in the applicable Service Level Agreement for each Service), you may request a Service credit equal to any charged amounts for such period.
- If you are a customer that is subject to the French Politique générale de sécurité des systèmes d’information de santé (PGSSI-S), you agree that your use of the Services complies with the PGSSI-S.

OpenAI Terms of Use

[Terms of use \(openai.com\)](https://openai.com/terms-of-use)

These Terms of Use apply when you use the services of OpenAI, L.L.C. thier affiliates, including their application programming interface, software, tools, developer services, data, documentation, and websites (“Services”). The Terms include their Service Terms, Sharing & Publication Policy, Usage Policies, and other documentation, guidelines, or policies they may provide in writing. By using their Services, you agree to those Terms. Their Privacy Policy explains how they collect and use personal information.

You must be at least 13 years old to use the Services. If you are under 18 you must have your parent or legal guardian’s permission to use the Services. If you use the Services on behalf of another person or entity, you must have the authority to accept the Terms on their behalf. You must provide accurate and complete information to register for an account. You may not make

STeMS Backend Services	Version: 2.0
Programmers Guide	Date: 7/24/2023
SWEN 670 Summer 2023 Team B	

your access credentials or account available to others outside your organization, and you are responsible for all activities that occur using your credentials.

These Terms take effect when you first use the Services and remain in effect until terminated. You may terminate these Terms at any time for any reason by discontinuing the use of the Services and Content. OpenAI may terminate these Terms for any reason by providing you at least 30 days' advance notice. OpenAI may terminate these Terms immediately upon notice to you if you materially breach certain sections of the agreement. if there are changes in relationships with third party technology providers outside of our control, or to comply with law or government requests. They may suspend your access to the Services if you do not comply with these Terms, if your use poses a security risk to them or any third party, or if they suspect that your use is fraudulent or could subject us or any third party to liability. Upon termination, you will stop using the Services and you will promptly return or, if instructed by OpenAI, destroy any Confidential Information.

- **Use of Services.** You may access, and OpenAI grants you a non-exclusive right to use, the Services in accordance with these Terms. You will comply with these Terms and all applicable laws when using the Services. OpenAI and their affiliates own all rights, title, and interest in and to the Services.
- **Feedback.** OpenAI appreciates feedback, comments, ideas, proposals and suggestions for improvements. If you provide any of these things, they may use it without restriction or compensation to you.
- **Restrictions.** You may not (i) use the Services in a way that infringes, misappropriates or violates any person's rights; (ii) reverse assemble, reverse compile, decompile, translate or otherwise attempt to discover the source code or underlying components of models, algorithms, and systems of the Services (except to the extent such restrictions are contrary to applicable law); (iii) use output from the Services to develop models that compete with OpenAI; (iv) except as permitted through the API, use any automated or programmatic method to extract data or output from the Services, including scraping, web harvesting, or web data extraction; (v) represent that output from the Services was human-generated when it is not or otherwise violate OpenAI's Usage Policies; (vi) buy, sell, or transfer API keys without our prior consent; or (vii), send OpenAI any personal information of children under 13 or the applicable age of digital consent. You will comply with any rate limits and other requirements in our documentation. You may use Services only in geographies currently supported by OpenAI.
- **Third Party Services.** Any third-party software, services, or other products you use in connection with the Services are subject to their own terms, and OpenAI is not responsible for third party products.