**Deployment & Operation Guide (Runbook)**

**For**

**Memory Magic App and Natural Language Understanding Module**

Version 1.2

Prepared by the Mesmerize Team and the Tongue Twisters Team:

Teresa Balbi, Karen Crumb, Daniel Avery, Sami Salim, Christian Cruz Jimenez, Kevin Bell, Andrew Rohn, Eskedar Endashw, Firehiwot Chari, Joseph Kalfus, Leelakrishnan Subramaniam, Malik Webster & Obinna Okonkwo.

October 21st, 2021

Table of Contents

[1 Introduction 4](#_Toc85802285)

[1.1 Purpose 4](#_Toc85802286)

[1.2 Intended Audience and Reading Suggestions 4](#_Toc85802287)

[1.3 Technical Project Stakeholders 4](#_Toc85802288)

[1.4 References 5](#_Toc85802289)

[1.5 Definitions, Acronyms, and Abbreviations 6](#_Toc85802290)

[2 Mobile Application 7](#_Toc85802291)

[2.1 Features, Packages, Plugins, and Widgets 7](#_Toc85802292)

[2.1.1 Features 7](#_Toc85802293)

[2.1.2 Packages 7](#_Toc85802294)

[2.1.3 Plugins 8](#_Toc85802295)

[2.1.4 Widgets 9](#_Toc85802296)

[3 Software Installation 10](#_Toc85802297)

[3.1 Android Studio 10](#_Toc85802298)

[3.2 Dart and Flutter 10](#_Toc85802299)

[3.3 Android Emulator 11](#_Toc85802300)

[3.4 Flutter and Dart plugins 12](#_Toc85802301)

[3.4.1 Test your development environment 13](#_Toc85802302)

[3.5 GitHub Desktop 13](#_Toc85802303)

[4 Prepare the Mobile Application for Use 13](#_Toc85802304)

[4.1 Cloning GitHub repository 13](#_Toc85802305)

[4.2 Run the Flutter Application 14](#_Toc85802306)

[5 Testing the Mobile Application 16](#_Toc85802307)

[5.1 Testing Objectives 16](#_Toc85802308)

[5.1.1 Unit Tests 16](#_Toc85802309)

[5.1.2 Integration Tests 20](#_Toc85802310)

[5.2 Testing Procedures 20](#_Toc85802311)

[6 Troubleshooting 21](#_Toc85802312)

[6.1 Emulator not responding 21](#_Toc85802313)

[6.2 Out of Memory Error 22](#_Toc85802314)

[6.3 Stuck at Running Gradle task 'assembleDebug'... 23](#_Toc85802315)

[6.4 Dependency Errors 23](#_Toc85802316)

[7 NLU Software Overview 24](#_Toc85802317)

[7.1 Amazon Lex 24](#_Toc85802318)

[7.2 Google BERT 24](#_Toc85802319)

[8 Software Installation 24](#_Toc85802320)

[8.1 Amazon Lex V2 24](#_Toc85802321)

[9 NLU Configurations 25](#_Toc85802322)

[9.1 Amazon Lex 25](#_Toc85802323)

[10 Testing the NLU Module 41](#_Toc85802324)

[10.1 Testing Objectives 41](#_Toc85802325)

[10.1.1 Unit Tests 41](#_Toc85802326)

[10.1.2 Integration Tests 43](#_Toc85802327)

[10.2 Testing Procedures 43](#_Toc85802328)

[11 Appendices 43](#_Toc85802329)

[11.1 Credits and Contributions 43](#_Toc85802330)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| September 16, 2021 | 1.0 | Initial Template | Teresa Balbi |
| October 20, 2021 | 1.1 | NLU Team Merge In | Tongue Twisters |
| October 21, 2021 | 1.2 | Minor Corrections | Mesmerize and Tongue Twisters Team |

# Introduction

## Purpose

The purpose of this Deployment and Operations Guide (Runbook) is to outline the necessary steps and illustrate details to install and deploy the Memory Magic App and the Natural Language Understanding (NLU) module.

## Intended Audience and Reading Suggestions

The intended audience for this guide is software developers, students, and technical stakeholders interested in installing/deploying the Memory Magic App and NLU module. The purpose of this document is to show each step, in detail, which any user can effortlessly follow.

Reading suggestions to assist the intended audience understand the systems used: *Flutter for Beginners*, by Alessandro Biessek and *Programming Flutter* by Carmine Zaccagnino.

## Technical Project Stakeholders

Table 1 shows the project stakeholders for the Memory Magic App and NLU module:

**Table 1**

Project Stakeholders

|  |  |
| --- | --- |
| Name | Role |
| Dr. Mir Assadullah | Stakeholder (Project Owner) |
| Roy Gordon | Stakeholder (Project Advisor) |
| Presley Muwan | Project Manager |
| Endalkachew Girma | Project Manager |
| App Team | |
| Teresa Balbi | Business Analyst |
| Karen Crumb | Lead Developer |
| Daniel Avery | Developer |
| Sami Salim | Developer |
| Christian Cruz Jimenez | Tester |
| Kevin Bell | UI/UX Designer |
| NLU Team | |
| Obinna Okonkwo | Project Manager |
| Andrew Rohn | Software Developer |
| Eskedar Endashw | Database Developer |
| Firehiwot Chari | Tester |
| Joseph Kalfus | System Analyst |
| Leelakrishnan Subramaniam | Software Developer |
| Malik Webster | Software Developer |

## References

Android studio. (n.d.). In *Developers*. Retrieved from https://developer.android.com/studio#downloads

Biessek, A. (2019). *Flutter for beginners: An introductory guide to building cross-platform mobile applications with Fuller and Dart 2*. Birmingham, UK: Packt Publishing Ltd.

Cloning a repository. (n.d.). In *GitHub Docs*. Retrieved from https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository

Dart. (n.d.). In *Dart.dev*. Retrieved from http://dart.dev

Flutter. (n.d.). In *Flutter.dev*. Retrieved from http://flutter.dev

Flutter App stuck at "Running Gradle task 'assembleDebug'... ". (2019, December 29). In *Stackoverflow*. Retrieved from https://stackoverflow.com/questions/59516408/flutter-app-stuck-at-running-gradle-task-assembledebug

Flutter favorites. (n.d.). In *Pub.dev*. Retrieved from https://pub.dev/

Flutter widget index. (n.d.). In *Flutter.dev*. Retrieved from https://flutter.dev/docs/reference/widgets

Install. (n.d.). In *Flutter.dev*. Retrieved from https://flutter.dev/docs/get-started/install

Introduction to widgets. (n.d.). In *Flutter.dev*. Retrieved from https://flutter.dev/docs/development/ui/widgets-intro

Ladd, S. (2015, December 1). mkustermann / flutter.github.io. In GitHub. Retrieved from https://github.com/mkustermann/flutter.github.io/blob/master/getting-started.md

Peter, C. (n.d.). Understanding and using Flutter packages. In *Educative*. Retrieved from https://www.educative.io/edpresso/understanding-and-using-flutter-packages

Set up an editor. (n.d.). In *Flutter.dev*. Retrieved from https://flutter.dev/docs/get-started/editor

Speech-to-Text. (n.d.). In *Google Cloud*. Retrieved from https://cloud.google.com/speech-to-text

Team Mesmerize. (n.d.). UMGC/fall2021. In *GitHub*. Retrieved from https://github.com/umgc/fall2021

Testing Flutter apps. (n.d.). In *Flutter.dev*. Retrieved from https://flutter.dev/docs/testing

Using packages. (n.d.). In *Flutter.dev*. Retrieved from https://flutter.dev/docs/development/packages-and-plugins/using-packages

Zaccagnino, C. (2020). *Programming Flutter: Native, cross-platform apps the easy way*. Raleigh, NC: The Pragmatic Bookshelf.

## Definitions, Acronyms, and Abbreviations

See the most used acronyms/abbreviations and their definitions below:

**Table 2**

Acronyms, Abbreviations, and Definitions

|  |  |
| --- | --- |
| Acronyms and Abbreviations | Definitions |
| AMD | Advanced Micro Devices |
| API | Application Program Interface |
| ASR | Automatic Speech Recognitions |
| AVD | Android Virtual Device |
| AWS | Amazon Web Services |
| BA | Business Analyst |
| Google BERT | Google’s Bidirectional Encoder Representations from Transformers |
| CPU | Central Processing Unit |
| DSO | DevSecOps |
| Flutter CLI | Flutter Command Line Tool |
| IAM | Identity and Access Management |
| IDE | Integrated Development Environments |
| iOS | iPhone operating system |
| JDK | Java Development Kit |
| JSON | JavaScript Object Notation |
| LEO | UMGC online platform class |
| NLP | Natural Language Processing |
| NLU | Natural Language Understanding |
| OS | Operating System |
| PM | Project Manager |
| RAM | Random-Access Memory |
| SDK | Software Development Kit |
| STML | Short Term Memory Loss |
| TTS | Text to Speech |
| UI | User Interface |
| UMGC | University of Maryland Global Campus |
| URL | Uniform Resource Locator |
| VM | Virtual Machine |

# Mobile Application

## Features, Packages, Plugins, and Widgets

### Features

* Simplicity – easy to navigate and keep user engaged. Clear, uncluttered screens, no ambiguity that leads to an easier and improved user experience.
* Speed - Fast loading screens that keep users engaged.
* Flexibility – Compatible with Android and iOS operating systems and can perform the necessary functions across different platforms.
* Security – Memory Magic Application deals with storing personal and sensitive information, so this feature will provide users the necessary security to deal with these types of information.
* Search Options – The Memory Magic App provide users the ability to search information throughout the application.
* Notifications – The Memory Magic App provides users relevant and personalized notifications.
* Internationalization / Multilingual – Multiple languages supported out of the box and extensible to add more support later.
* Internationalization/Multilingual – Multiple languages supported out of the box and extensible to add more support later.

See below specific features used in the Memory Magic App:

* NLU Interaction - app integrates with an amazon Lex Natural Language Understanding system.
* Resource-permission- System explicitly request access Permission to the users’ device resources.
* Activate and deactivate listening mode on button click.
* Speech-to-Text Transcribe: System can transcribe user voice to text in multiple languages.
* Note Management: User have the ability to create, delete, edit and search notes.

### Packages

Packages can be defined as “shared packages contributed by other developers to the Flutter and Dart ecosystems. This allows developers to quickly build an app without having to developing everything from scratch” (Peter, n.d., para. 2).

Packages are published into pub.dev. This webpage contains packages and dependencies that are compatible with Flutter. See Table 3 for list of core packages.

#### Installing a package dependency into an app

1. Dependencies
   1. Open the pubspec.yaml file located inside the app folder and add a specific package under dependencies.
2. Install
   1. From the terminal: Run flutter pub get.  
      OR
   2. From Android Studio/IntelliJ: Click Pub Get in the action ribbon at the top of pubspec.yaml.
   3. From VS Code: Click “Get Packages” located in right side of the action ribbon at the top of pubspec.yaml.
3. Import
   1. Add a corresponding import statement in the Dart code.

Table 3 shows the core packages used in the Memory Magic App.

**Table 3**

*Packages and Descriptions*

|  |  |
| --- | --- |
| Packages | Description |
| flutter\_localizations | Package dependency used to set up a localization |
| speech\_to\_text | Package that exposes device specific speech to text recognition capability |
| encrypt | A set of high-level APIs over PointyCastle for two-way cryptography |
| timeago | A library useful for creating fuzzy timestamps |
| flutter\_tts | Package for Text to Speech |
| intl | Package that contains code to deal with internationalized/localized messages, date and number formatting and parsing, bi-directional text, and other internationalization issues |
| flutter\_mobx | Package that provides a set of Observer widgets that automatically rebuild when the tracked observables change |
| http | A composable, multi-platform, Future-based API for HTTP requests |

*Note.* See https://github.com/umgc/fall2021/blob/developer/pubspec.yaml for a complete list of packages.

### Plugins

The installation of Flutter and Dart plugins instructions vary by platform:

* MacOS installation:

Use the following instructions for macOS:

1. Start Android Studio.
2. Open plugin preferences (Preferences > Plugins).
3. Select the Flutter plugin and click Install.
4. When prompted click Yes to install the Dart plugin.
5. When prompted click Restart.

* Windows Installation:

Use the following instructions for Windows:

1. Open plugin preferences (File > Settings > Plugins).
2. Select Marketplace, select the Flutter plugin and click Install.
3. Read/accept privacy notice.
4. When prompted, click Yes to install plugins
5. Once installation is completed, click Restart for changes to take effect.

### Widgets

Flutter is a modern framework for cross platform mobile development. “Widgets describe what their view should look like given their current configuration and state. When a widget’s state changes, the widget rebuilds its description, which the framework diffs against the previous description [to] determine the minimal changes needed” (Introduction to widgets, n.d., para. 1).

Table 4 shows the core widgets used in the Memory Magic App.

**Table 4**

*Widgets and Descriptions*

|  |  |
| --- | --- |
| Widgets | Description |
| Text | This widget lets you create a run of styled text within your application. |
| Container | This widget lets you create a rectangular visual element. |
| ElevatedButton | A filled button whose material elevates when pressed. |
| Row, Column | These flex widgets let you create flexible layouts in both the horizontal (Row) and vertical (Column) directions. |
| Padding | This widget that insets its child by the given padding. |
| Image | This widget that displays an image. |
| Scaffold | Implements the basic Material Design visual layout structure. This class provides APIs for showing drawers, snack bars, and bottom sheets. |
| Table | A widget that uses the table layout algorithm for its children. |
| DropdownButton | Shows the currently selected item and an arrow that opens a menu for selecting another item. |
| Align | This widget aligns its child within itself and optionally sizes itself based on the child's size. |

*Note.* See https://flutter.dev/docs/reference/widgets for a complete list of widgets.

# Software Installation

These instructions will guide the user to set up the development environment in Windows and MacOS operating systems. Some steps could be different from one system to the other, but this section will provide the fundamental steps to help the user to identify the correct tools. For the full step-by-step instructions, please refer to the official web pages provided below for each one of the systems.

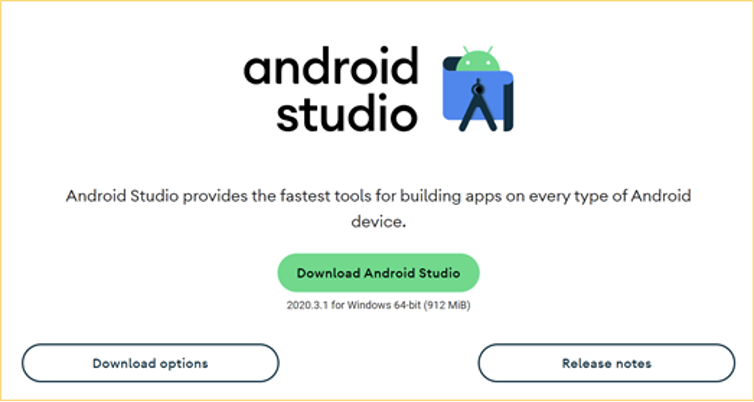
## Android Studio

Download and Installation

1. Navigate to https://developer.android.com/studio and download the latest version of Android Studio for your Operating System (OS).

**Figure 1**

*Android Studio*



*Note:* Download page for Android Studio. From Android studio, by Developers, n.d. (https://developer.android.com/studio).

1. To install Android Studio on your computer, follow the official instructions for your OS at https://developer.android.com/studio/install
2. After installing Android Studio successfully, you can continue installing the other tools.

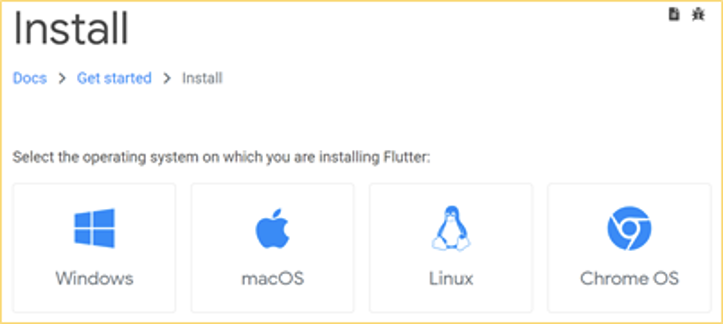
## Dart and Flutter

Download and Installation

1. To download Flutter, visit the official website <https://flutter.dev/>, select the correct OS and follow the installation instructions for your OS.

**Figure 2**

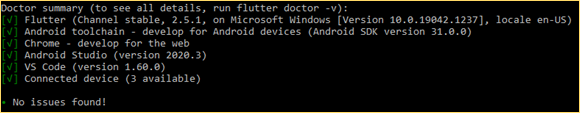
*Install*



*Note: Get started section on flutter.dev. From Install, by Flutter.dev, n.d. (*[*https://flutter.dev/docs/get-started/install*](https://flutter.dev/docs/get-started/install)*).*

1. After following the official installation instructions, make sure to run flutter doctor to verify that your installation has been completed correctly and that Flutter has located your Android Studio installation. Everything should be green and no issues found.

**Figure 3**

*Doctor Summary*  


* 1. \* If Flutter can’t locate your Android Studio, run flutter config --android-studio-dir <android studio directory here> to indicate the location of Android Studio.

1. After successfully installing Android Studio and Flutter SDK, you can continue to the next step to set up your Android Studio emulator.

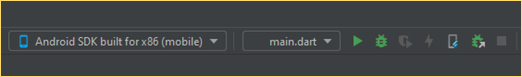
## Android Emulator

Set up your Android Emulator

1. To set up your emulator correctly, please follow the official instructions for your OS on the Flutter web page.
2. Windows instructions can be found at https://flutter.dev/docs/get-started/install/windows
3. macOS instructions can be found at https://flutter.dev/docs/get-started/install/macos
4. After successfully installing your emulator, you should be able to see it integrated into your Android Studio and you should be able to run it.

**Figure 4**

*Android SDK*





1. After installing Flutter, Android Studio, and setting up your emulator, you must agree to the licenses of the Android SDK platform to use the development environment correctly. Open your console terminal and run the command flutter doctor --android-licenses. Make sure that you have installed all other additional requirements for your OS.

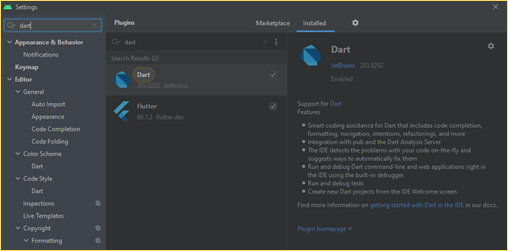
## Flutter and Dart plugins

Install Flutter and Dart plugins in Android Studio

1. To install the correct plugins in Android Studio, please follow the official instructions from https://flutter.dev/docs/get-started/editor?tab=androidstudio.

**Figure 5**

*Dart Plugin Install*



1. After installing the correct plugins your environment should be ready to use.

### Test your development environment

Test your development tools

1. To test your development environment, you can create a simple Flutter application in your Android Studio: Create New Flutter Project.  
   \* Make sure that your Flutter SDK path reflects the specific SDK location.
2. After Android Studio has created the Flutter project you can run the application in your emulator without problems.
3. For more instructions to run a test drive, please follow the official instructions from https://flutter.dev/docs/get-started/test-drive?tab=androidstudio.

## GitHub Desktop

This tool is not required for the development of the application but for people with less experience using GitHub with command lines, it would be beneficial to download the desktop version. GitHub will be used as the application code repository.

Download and Install GitHub Desktop

1. Visit the official website to download the application at https://desktop.github.com/
2. After the download, install the application and sync your GitHub account with GitHub Desktop, or create a new GitHub account.
3. After setting your account you should be able to access the application repository and collaborate on the code.

# Prepare the Mobile Application for Use

## Cloning GitHub repository

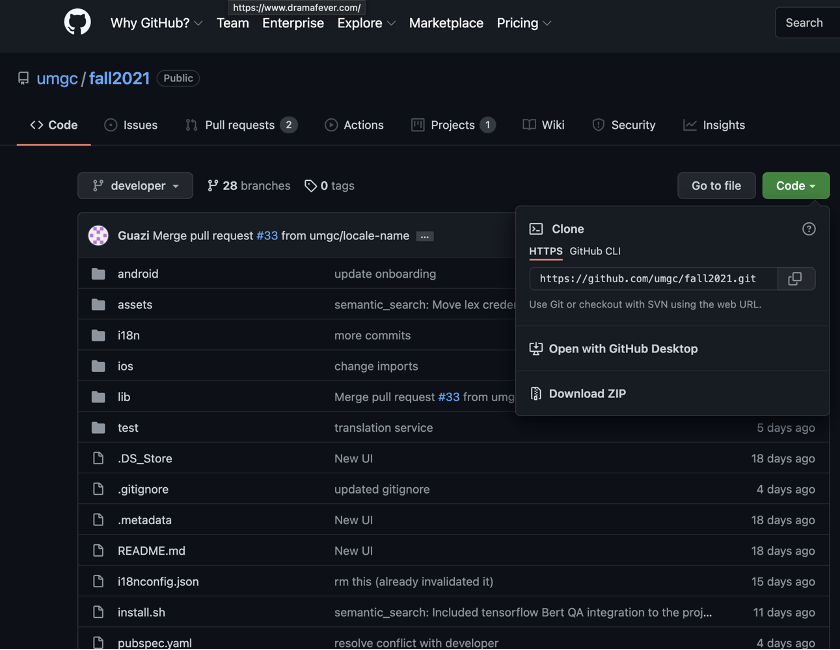
Cloning a GitHub repository to a local computer makes it easier to fix conflicts, add/remove files, and push commits.

1. Open the command line terminal.
2. Go to your home directory.
3. Type the command git clone https://github.com/umgc/fall2021.git
4. Or you can navigate through the website:
   1. Navigate to https://github.com/umgc/fall2021
   2. Click on “Code”
   3. To clone the code using HTTPS, click on HTTPS and download ZIP

Figure 6 shows the UMGC Fall 2021 GitHub page.

**Figure 6**

*Clone GitHub*



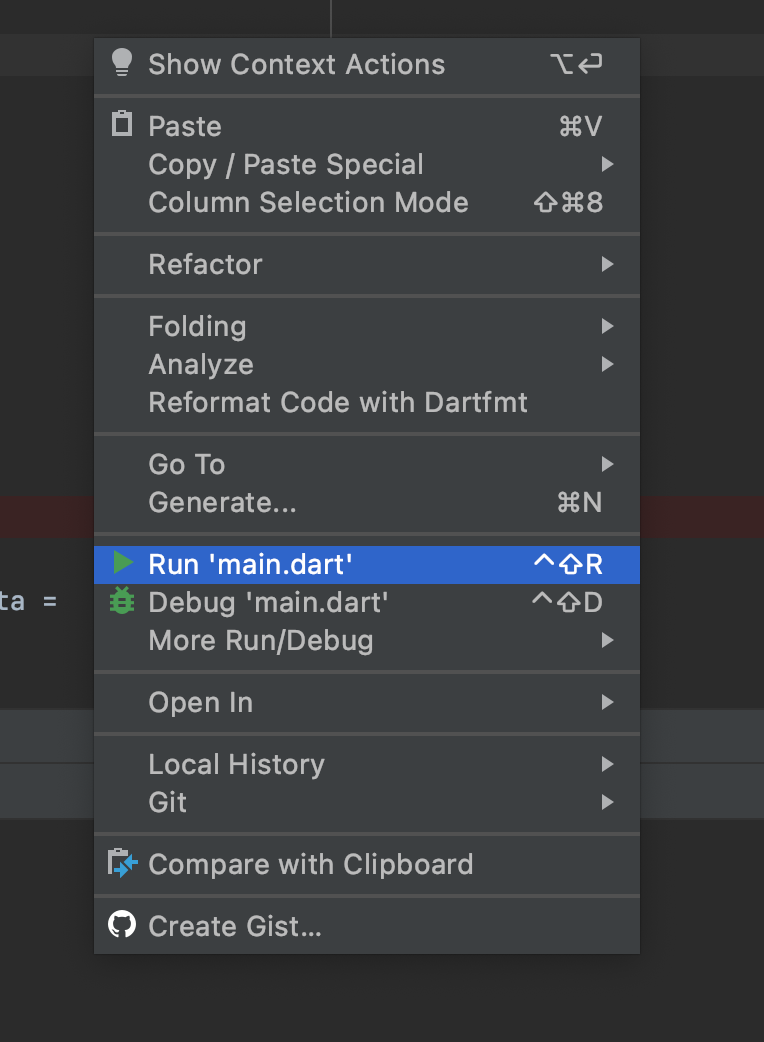
*Note*: Clone Code section on GitHub. From Team Mesmerize, by GitHub, n.d. (<https://github.com/umgc/fall2021>).

## Run the Flutter Application

To run Flutter applications on a device, first start the device on the emulator then open the file main.dart in Android Studio IDE, then right click and choose the option “Run main.dart”

**Figure 7**

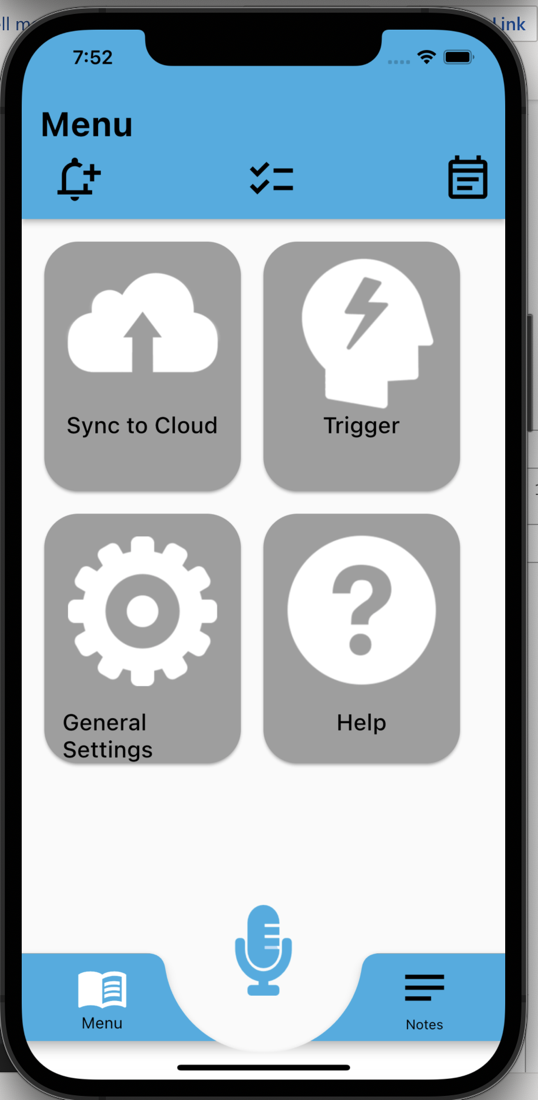
*Run main.dart*



The starter app opens on the device, as shown on Figure 8.

**Figure 8**

*Flutter App*



# Testing the Mobile Application

This section examines the testing process for the Memory Magic Application. Testing the Memory Magic Application is necessary to ensure that all the required deliverables have been successfully implemented and to provide quality control and quality assurance. Several full-system, end-to-end (E2E) smoke tests shall be created for common use cases to ensure intended functionality and to serve as a safeguard in the case of regression of a feature that has already been integrated into the system. Static analysis shall be completed on the application to ensure issues may be rectified before ultimate delivery of the application. This includes features that may be available on GitHub, and/or any necessary third-party tools ensuring the code is free of any major errors/bad syntax that may lead to exploits. Table 5 shows the Test Matrix for the General Tests and Table 6 shows the Test Matrix for Integration Tests. The Test Matrix will be utilized to confirm that all features are working as intended. The tests will ensure that the UMGC Software Capstone Project (SWEN 670, Fall 2021) will be successful.

## Testing Objectives

### Unit Tests

A unit test assesses a single function, method, or class. The goal is to verify the accuracy of a logical unit.

**Table 5**

Memory Magic App Unit Tests

|  |  |  |
| --- | --- | --- |
| General Tests: | iPhone | Android |
| 1. Application Opens Up  Success case: Wait for the “Speak” button to appear which means that the application has successfully opened. |  |  |
| 2. From any page, click Calendar  Success case: Wait for anticipated elements to appear. |  |  |
| 3. From any page, click on Settings  Success case: Wait for anticipated elements to appear. |  |  |
| 4. From any page, click on Notes  Success case: Wait for anticipated elements to appear. |  |  |
| 5. From any page, click on Home  Success case: Wait for anticipated elements to appear. |  |  |
| 6. From any page, click on Bell  Success case: Wait for anticipated elements to appear. |  |  |
| 7. From any page, click on Mic  Success case: Wait for anticipated elements to appear. |  |  |
| Home Screen: | iPhone | Android |
| 8. Click on “Speak”  Success case: Mic screen will appear. |  |  |
| 9. Click on “Text”  Success case: Enter a note by text screen will appear. |  |  |
| Mic Screen: | iPhone | Android |
| 10. Click on “Press to Record (Mic)”  Success case: Speak dialog appears. |  |  |
| Menu Screen:  11. Click on sync to cloud  Success case: Enter state for enabling sync to cloud. |  |  |
| 12. Click on trigger  Success case: Wait for anticipated elements to appear. |  |  |
| 13. Click on general setting  Success case: Wait for anticipated elements to appear. |  |  |
| 14. Click on help  Success case: Wait for anticipated elements to appear. |  |  |
| Note Screen: | iPhone | Android |
| 15. Search for note – no results  Success case: No results appear. |  |  |
| 16. Search for a note - results  Success case: Searched for note will appear. |  |  |
| 17. Modify text after search note  Success case: Note is updated. |  |  |
| 18. Click on “Add A Note”  Success case: Wait for anticipated elements to appear. |  |  |
| Delete a Note Screen | iPhone | Android |
| 19. Click on Delete Note  Success case: Note is deleted from datastore. |  |  |
| 20. Click on Cancel. |  |  |
| Edit a note Screen | iPhone | Android |
| 21. Click on Save Changes  Success case: Note is saved in datastore. |  |  |
| 22. Click on Cancel Changes  Success case: Note is not saved in datastore. |  |  |
| Trigger Screen | iPhone | Android |
| 23. Change all inputs and save  Success case: Triggers are saved in datastore. |  |  |
| 24. Change all inputs and cancel  Success case: Triggers are not saved in datastore. |  |  |
| Sync to Cloud Screen | iPhone | Android |
| 25. Mock login to google cloud  Success case: After mock response, user is linked to google cloud. |  |  |
| General Setting Screen | iPhone | Android |
| 26. Days to keep notes and save  Success case: Settings are updated for the user. |  |  |
| 27. Change font size notes  Success case: Settings are updated for the user. |  |  |
| 28. Change font size menu  Success case: Settings are updated for the user. |  |  |
| 29. Change language from English to Arabic  Success case: Settings are updated for the user. |  |  |
| 30. Change language from English to Arabic back to English, then save  Success case: Settings are updated for the user. |  |  |
| Security Setting Screen | iPhone | Android |
| 31. Setup fingerprint  Success case: Settings are updated for the user. |  |  |
| 32. Reset fingerprint  Success case: Settings are updated for the user. |  |  |
| 33. Press Cancel Button.  Success case: State is unchanged. |  |  |
| Calendar Screen | iPhone | Android |
| 34. Click a calendar event  Success case: Wait for anticipated elements to appear. |  |  |

### Integration Tests

An integration test assesses the application as a whole or a large part of it. The goal is to verify that all features tested can work together as expected.

**Table 6**

Integration Tests

|  |  |  |
| --- | --- | --- |
| Integration Tests | iPhone | Android |
| \*\*Integration with NLU (Team B): |  |  |
| 1. Application handles successful response from NLU API.  Success case: Obtain desired response. |  |  |
| 2. Application handles all non-success responses from NLU API.  Success case: Error is handled gracefully. |  |  |
| Integration with Google Drive: | iPhone | Android |
| 3. Application handles successful link with Google API  Success case: Profile is linked. |  |  |
| 4. Application handles unsuccessful link with Google API  Success case: Error is handled gracefully. |  |  |

## Testing Procedures

1. Choose Windows (version 10 or later) operating system (64-bit with a x64 based processor

laptop) or iOS operating system.

1. Install Microsoft Teams – to be used in collaboration with TTT team, DevOps and team Mesmerize members.
2. Install Flutter version 2.5.
3. Install Android Studio version 4.3 with Flutter and Dart plugins.
4. Install GitHub - to be used in collaboration with TTT team, DevOps and team Mesmerize

members.

1. Clone the GitHub repository https://github.com/umgc/fall2021
2. Execute the application
3. Manually test each feature of the application
4. Document the test results
5. If any of the required tests fail, fix the issues, and collaborate with NLU team, DevOps and smartphone app team members
6. Rerun the updated failed tests
7. Complete the test results document
8. Complete the test report documentation

# Troubleshooting

After installing and using your development environment you can still find some issues with using Flutter and Android Studio. This section will cover some of the most common issues and a quick possible troubleshooting solution to the problem. Some of the troubleshooting steps provided in this section could be different depending on the OS you are using and depending on the application versions used. For more specific solutions to your problem, please visit the official website of Flutter at https://flutter.dev/docs and Android Studios at https://developer.android.com/studio/troubleshoot.

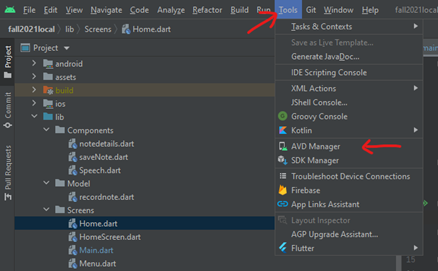
## Emulator not responding

After installing the emulator, there may be some issues when trying to start the emulator or after using it multiple times. If the emulator is unable to execute and/or run properly depending on the configuration and resources being used from the local/host machine. A possible solution is to reset the emulator data and return it to the same state as when it was first defined:  
 \* To perform these steps your emulator should not be running.

1. If you have a Project already open on Android Studio, click on Tools > ADV Manager, as shown on Figure 9.

**Figure 9**

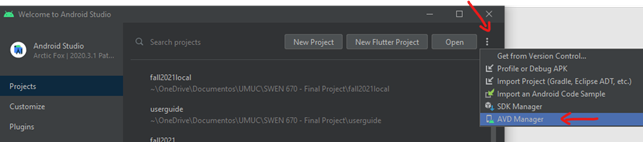
*Tools*



* 1. On the Welcome page of Android Studio, click on the three dots at the top right corner > ADV Manager, as shown on Figure 10.

**Figure 10**

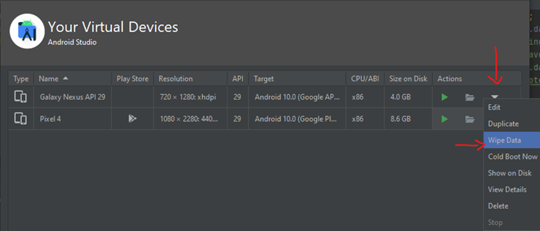
*ADV Manager*



1. After opening the ADV Manager, identified the emulator you want to use and click on the dropdown arrow > Wipe Data, as shown on Figure 11.

**Figure 11**

*Your Virtual Devices*



1. After wiping your device data, you can perform a ‘Cold Boot’ from the same location you just wiped the data to restart the emulator device. Click on the dropdown arrow > Cold Boot Now.

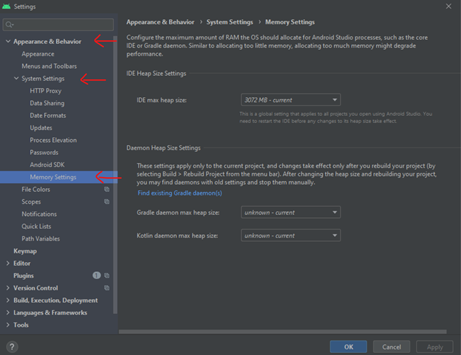
## Out of Memory Error

This error is very common if the local machine you are using to run the emulator does not have sufficient resources to handle the virtual device. A common resolution to this problem is to close all unnecessary applications running in the background of your computer to free up virtual memory. Also, you should consider increasing the assigned memory to your virtual emulator by following these steps:

1. On Android Studio click on File > Settings to open the general settings.
2. On the general settings screen click on Appearance & Behavior > System Setting > Memory Settings, as shown on Figure 12.

**Figure 12**

*Your Virtual Devices*



1. Here you can customize the amount of memory you want to assign to your environment. The amount you assign should be based on the resources of your computer but is recommended to have at least 2,048 MB assigned.
2. After assigning your memory you can click Apply > Ok to apply and save the changes.
3. Restart Android Studio.
4. More information can be found at the official web page of Android Studio at https://developer.android.com/topic/performance/memory.

## Stuck at Running Gradle task 'assembleDebug'...

This is a common issue that the user could present when trying to start the app in the emulator. If you receive the message “Running Gradle task 'assembleDebug'… (This is taking an unexpectedly long time.)” you can consider the following troubleshooting:

1. Open your terminal app/console.
2. Navigate to the directory where you have the Flutter project.
3. On your Flutter project directory run this command ./gradlew clean.
4. You can build the Gradle from the directory or let Android Studio build it when you run the app. To build Gradle from the directory run ./gradlew build. Also, you can combine both commands to clean and build the Gradle by running ./gradlew clean build (Koderstory, 2020).

## Dependency Errors

Most of the time you could get unexpected dependency errors on your code. This normally occurs when multiple versions of a library get imported to the project and could cause consistency issues between the versions. A common resolution to this problem could be to clear the Flutter built files and reimport the libraries. To do so follow these steps:

1. Open your terminal app/console.
2. Navigate to the root directory of your Flutter project.
3. Run the command flutter clean to remove all the dependencies.
4. Now reimport the newest libraries to reimport the correct dependencies by running flutter pub get.

# NLU Software Overview

## Amazon Lex

Amazon Lex is a cloud based natural language processing service. For the NLU module, this is a main component for understanding a NLU user’s query (in this case, what the smartphone app passes in), and returning either follow up questions, or the response to the smartphone app.

## Google BERT

Google BERT will be using Tensor Flow Lite to act as a model to aid Amazon Lex in queries. Whereas Amazon Lex determines the *intent* of the app request, BERT will query previously stored notes based on the intent and attempt to answer queries based on what it finds in the datastore.

📁 The Google BERT model is added as a precompiled binary into the GitHub repository for the smartphone app to call as the NLU Service.

# Software Installation

## Amazon Lex V2

Amazon Lex V2 is an AWS service for building conversational interfaces for applications using voice and text. Amazon Lex V2 provides the functionality and flexibility of natural language understanding (NLU) and automatic speech recognition (ASR) so you can build highly engaging user experiences with lifelike, conversational interactions, and create new categories of products.

Amazon Lex V2 provides API operations that can integrate with the existing applications. Before using Amazon Lex V2 for the first time, complete the following tasks:

Sign Up for AWS

To create an AWS account

* Open https://portal.aws.amazon.com/billing/signup
* Follow the online instructions.
  + Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.
  + Write down your AWS account ID because you'll need it for the next task.
* Create an IAM User

Services in AWS, such as Amazon Lex V2, require that you provide credentials when you access them so that the service can determine whether you have permissions to access the resources owned by that service. The console requires password. You can create access keys for your AWS account to access the AWS CLI or API.

However, AWS don't recommend that you access AWS using the credentials for your AWS account. Instead, they recommend that you:

* Use AWS Identity and Access Management (IAM) to create an IAM user
* Add the user to an IAM group with administrative permissions
* Grant administrative permissions to the IAM user that you created.

You can then access AWS using a special URL and the IAM user's credentials. Refer this URL if you need help https://docs.aws.amazon.com/lexv2/latest/dg/gs-account.html.

# NLU Configurations

## Amazon Lex

To configure Amazon Lex correctly, you must train the language model on the different *intents* and *utterances* that your application will need to understand, answer, or accomplish. Please see the Software Installation section within this document to see how to access the Amazon Lex console.

Within the Amazon Web Service Lex console, you must first add in the different **slot types**. Each entry is delimited by a semi-colon. The etc abbreviation at the end of some entries means that other entries can be included, and the list below is not an all-inclusive listing.

📁 The JSON is posted in the GitHub repository, for easier configuration.

Below, you will find a Table 7 with the Slot Types and values:

**Table 7**

Amazon Lex Slot Type Value Configuration

|  |  |  |
| --- | --- | --- |
| **Slot Types** | **Slot Type Values** | **\*\*Restricted to Slot Values** |
| SubjectType | I; she; you; we; they; he; it; i; the plumber; etc |  |
| RecurringType (\*\* restrict the **Slot value resolution** to **Restrict to slot values**) | Hourly | Hourly, every hour, each hour, once an hour, once every hour; once per hour; one time an hour; etc |
| Daily | Daily; every day; each day; once a day; once every day; once each day; once per day; one time a day; etc |
| Weekly | Weekly; every week; each week; once a week; once each week; once every week; once per week; one time a week; once time each week; one time every week; one time per week |
| Biweekly | Biweekly; twice a month; twice each month; twice every month; twice per month; every two weeks; once every two weeks; two times a month; two times each month; etc |
| Monthly | Monthly; every month; each month; once a month; once each month; once every month; once per month; one time a month; one time each month; etc |
| Quarterly | Quarterly; every quarter; each quarter; once a quarter; once per quarter; one time a quarter; one time each quarter; one time every quarter; one time per quarter; every three months; once every three months; four times a year; etc |
| Semi-annual | Semi-annual; semi annual; semiannual; semi-annually; semi annually; semiannually; twice a year; twice each year; two times a year; two times each year; two times per year; etc |
| Annual | Annual; annually; yearly; every year; each year; once a year; once each year; once every year; once per year; one time a year; one time each year; etc |
| AuxiliaryVerbType | Is; are; was; were; did; shall; will; should; would; may; might; must; can; could; does; do; ought to; have to; had better; had better be able to; got to; want to; would like to; am able to; am being; etc |  |
| BeType | Is a; was a; will be a; has been a; is going to be a; was going to be a; is an; was an; will be an; has been an; is going to be an; was going to be an; etc |
| Have Type | Have a; has a; had a; will have a; should have a; have an; has an; had an; will have an; should have an; etc |
| EventType | Root canal; teeth whitening; doctor appointment; dentist appointment; meeting; birthday; anniversary; etc |
| ActionEventType | Buy eggs; pick up milk; meet Tom; bring sandwiches; finish the assignment; plan the wedding; work on the project; look for eggs; etc |
| QuestionType | Who; what; when; where; why; how; which; for what; for whom; for which; etc |
| QuestionAuxilliaryVerbType | Am; is; are; was; were; being; has; have; had; did; shall; will; should; would; may; might; must; can; could; does; do; ought |

Next, within the Amazon Web Service Lex console, you must add in the different **sample utterances**. Each entry is delimited by a semi-colon. The etc abbreviation at the end of some entries means that other entries can be included, and the list below is not an all-inclusive listing. Below, you will find a Table 8 with the different sample utterances:

**Table 8**

Amazon Lex Sample Utterance Configuration

|  |  |
| --- | --- |
| Description | Sample Utterances |
| AppHelpCalendar | how do I use the calendar; how do I see events on the calendar; how do I check events on the calendar; how do I add things to the calendar; how do I remove events from my schedule; how do I delete things from my agenda; I need help with the calendar; |
| AppHelpChecklist | how do I use the checklist; how do I create a to do list; how do I check off things on the checklist; how do I add items to the checklist; how do I remove stuff from my to do list; how do I delete items on the checklist; I need help with the checklist; how do I update my to do list; |
| AppHelpCloud | how do I upload stuff to the cloud; how do I download things from the cloud; how do I sync files with the cloud; how do I use the cloud service; how do I enable the cloud; how do I disable the cloud feature; how do I set up the cloud; how do I configure the cloud settings; how do I turn on the cloud; how do I turn off the cloud feature; how do I start the cloud service; how do I stop using the cloud; how do I back up files; how do I restore files; how do I save nates to the cloud; I need help with the cloud settings; |
| AppHelpFont | how do I change the font; how do I enlarge the font; how do I decrease the font size; how do I increase the font size; how do I make the text bigger; how do I make the text smaller; how do I select a different font type; how do I change font style; how do I make the text bold; how do I change the text color; how do I choose a different font; I need help with the font; I need help with the text; |
| AppHelpLanguage | how do I change the language; how do I pick another language; how do I update the language setting; how do I change the dialect; I need help with language |
| AppHelpMic | how do I enable the mic; how do I turn the microphone on; how do I disable the microphone; how do I turn off the mic; how do I adjust the mic; how do I calibrate the mic; how do I use the microphone; how do I stop the mic; how do I start the microphone; why can I not use the mic; why is the microphone not working; how do I set up the mic; how do I mute the mic; how do I unmute the microphone; I need help with the mic; |
| AppHelpNotes | how do I create a note; how do I delete note; how can I update notes; how do I edit notes; how do I remove a note; how do I make notes; I need help with my notes; |
| AppHelpNotifications | how do I clear notifications; how do I delete notifications; how do I remove notifications; how do I change notification sound; how do I disable notifications; how do I enable notifications; how do I enable alerts; how do I disable alerts; I need help with notifications; |
| AppHelpSecurity | how do I enable the fingerprint scanner; how do I turn the fingerprint sensor on; how do I disable the fingerprint scanner; how do I turn off the fingerprint detector; how do I adjust the security settings; how do I update my security options; how do I use the fingerprint scanner; how do I stop the fingerprint sensor; how do I start the fingerprint scanner; why can I not use the fingerprint sensor; why is my fingerprint not working; how do I set up the fingerprint scanner; I need help with the fingerprint sensor; I need help with the security settings; |
| AppHelpSound | how do I increase the volume; how do I decrease the volume; how do I change the sound; how do I mute sound; how do I unmute sound; how do I change sounds; how do I update sounds; how do I turn on noise; how do I turn off noise; how do I enable sound; how do I disable sound; how do I toggle noise; how do I update volume; how do I mute the device; how do I unmute the device; I need help with the sound; I need help with volume; |
| AppHelpTheme | how do I change the theme; how do I choose a different theme; I need help changing the theme; how do I change the color; how do I update the color; how do I pick another color; how do I select another theme; I need help with the theme; |
| AppHelpTrigger | how do I create a trigger phrase; how do I delete a trigger word; how do I update a trigger; how do I make a trigger; how do I edit a trigger word; how do I change a trigger phrase; how do I enable a trigger; how do I disable a trigger; how do I turn on a trigger word; how do I turn off a trigger phrase; how do I add a trigger; how do I remove a trigger phrase; how do I set up a trigger; I need help with triggers; |
| AppNavCalendar | take me to the calendar screen; return to the calendar page; I want to see the calendar screen; I want to adjust my calendar; I need to update my calendar; direct me to the calendar; load the calendar section; go back to the calendar page; I want to go to the calendar screen; where is the calendar; show me the calendar page; I want to see my events; I would like to see my agenda; I want to look at my events; I would like to check the calendar; show me my events; what events do I have on my schedule; I want to see my schedule; show me my schedule; I would like to see my itinerary; |
| AppNavChecklist | take me to checklist screen; return to checklist page; I want to see to do screen; I would like to adjust my checklist; I need to update the checklist; direct me to my checklist; load the to do section; go back to the checklist page; I want to go to the to do list screen; where is my checklist; show the to do page; I want to see the checklist; I would like to see the to do list; I want to look at the checklist; I would like to check my checklist; show me the checklist; what is on the checklist; I want to see the to do list; I would like to see the checklist; I would like to create a checklist; I would like to edit my to do list; I would like to update my checklist; I would like to add a checklist; I want to adjust my to do list; |
| AppNavCloud | take me to the cloud screen; return to the cloud page; I want to upload to the cloud; I want to download from the cloud; I want to sync with the cloud; direct me to the cloud section; load the cloud screen; go back to the cloud page; I want to go to the cloud screen; where is the cloud; show me the cloud section; I would like to connect to the cloud; I would like to upload notes; I want to download my old notes; I want to get my old notes from the cloud; |
| AppNavHelp | take me to the help screen; I need help with the app; return to the help page; I want to see the help screen; direct me to the help options; load the help menu; go back to the help section; I want to go to the help page; where can I get help; show me the help section; take me to the support page; show me the support section; where are the support options; I require assistance with some features; I need help with a feature; |
| AppNavMenu | take me to the main screen; return the main menu; I want to see the menu; direct me to the home screen; load the home page; go back to the main page; go to the first screen; I want to go to the main page; show me the menu; |
| AppNavMic | take me to the mic screen; return to the mic screen; I want to use the mic; direct me to the mic; load the mic page; go back to the mic; I want to go to the mic page; where is the mic; show me the mic screen; I would like to ask a question; I would like to record a note; I want to say something; I want to speak to the app; I want to record something; |
| AppNavNotes | take me to the notes screen; return to the notes page; I want to see my notes; direct me to the notes screen; load the notes page; go back to the notes screen; I want to go to the notes screen; where are my notes; why can I not see my notes; where are the notes; show me my notes; I want to delete a note; I want to edit a note; I want to remove a note; I would like to update a note; I would like to create a note; I want to add a note; I want to make a note; |
| AppNavNotifications | take me to the notifications screen; return to the notifications page; I want to see my notifications; direct me to my notifications; load my notifications; load the notification screen; go back to the notifications page; I want to go to the notifications screen; where are my notifications; show me my notifications; take me to the alerts page; where are my alerts; load the alerts screen; I want to clear my notifications; I would like to check my notifications; I want to see my alerts; |
| AppNavSecurity | take me to the security settings screen; return to the security page; I want to see the security screen; I want to adjust my security settings; I need to update my security settings; direct me to the security options; load the security section; go back to the security settings page; I want to go to the security screen; where are the security options; show me the security menu; how do I change the security settings; I want to enable the fingerprint scanner; I would like to disable the fingerprint sensor; I want to use my fingerprint to access notes; I would like to scan my fingerprint again; |
| AppNavSettings | take me to the settings menu; return to the settings page; I want to see the settings screen; I want to adjust my settings; I need to update my settings; direct me to the settings menu; load the settings section; go back to the settings page; I want to go to the settings screen; where are the settings; show me the settings menu; how do I change settings; take me to the general settings screen; I want to change the language; I would like to choose a different language; I want to update the language; I want to make the text bigger; I would like to change the font size; I want to change the font color; I would like to pick a different text color; I want to set when notes will be deleted; I would like to determine when notes shall be deleted; I want to change the font style; |
| AppNavTrigger | take me to the trigger screen; return to the trigger page; I want to set a trigger phrase; I need to update my trigger; I want to see my trigger word; direct me to the trigger section; load the trigger page; go back to the trigger screen; I want to go to the trigger screen; where is my trigger; show me my trigger; I want to change my trigger; I would like to record a trigger; I want to store a trigger; I would like to save a trigger sentence; |
| Compliment | you are the best; you are smart; you are very helpful; you are reliable; I like your voice; you are so thoughtful; you are a great listener; I like your style; you are so reliable; what would I do without you; that is very kind of you; I like you; I think you are awesome; |
| CreateActionEvent | {SubjectType} {AuxiliaryVerbType} do something; {SubjectType} {AuxiliaryVerbType} do something later; later {SubjectType} {AuxiliaryVerbType} do something; {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} on {Date}; on {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} at {Time}; at {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {Date}; {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {Date} {Time}; {Date} {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; set a reminder that {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {ActionEventType}; {ActionEventType} by {Time}; by {Time} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} before {Time}; before {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} sometime {Date}; sometime {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} later {Date}; later {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} earlier {Date}; earlier {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {ActionEventType} {Date}; {Date} {ActionEventType}; {ActionEventType} on {Date}; on {Date} {ActionEventType}; {ActionEventType} after {Time}; after {Time} {ActionEventType}; {ActionEventType} after {Date}; after {Date} {ActionEventType}; {ActionEventType} before {Time}; before {Time} {ActionEventType}; {ActionEventType} before {Date}; before {Date} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} after {Date}; after {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; remind me to {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} at {Time} {Date}; at {Time} {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} by {Date}; by {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} by {Time}; by {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; make a note that {SubjectType} {AuxiliaryVerbType} {ActionEventType}; create a note that {SubjectType} {AuxiliaryVerbType} {ActionEventType}; add to my calendar that {SubjectType} {AuxiliaryVerbType} {ActionEventType}; |
| CreateEvent | {SubjectType} {HaveType} {EventType}; there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} on {Date}; on {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} on {Date}; on {Date} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} at {Time}; at {Time} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} at {Time}; at {Time} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} {Date}; {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} {Date}; {Date} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} {Date} {Time}; {Date} {Time} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} {Date} {Time}; {Date} {Time} there {BeType} {EventType}; set a reminder that {SubjectType} {HaveType} {EventType}; set a reminder that there {BeType} {EventType}; {EventType}; {EventType} by {Time}; by {Time} {EventType}; {SubjectType} {HaveType} {EventType} before {Time}; before {Time} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} before {Time}; before {Time} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} sometime {Date}; sometime {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} sometime {Date}; sometime {Date} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} later {Date}; later {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} later {Date}; later {Date} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} earlier {Date}; earlier {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} earlier {Date}; earlier {Date} there {BeType} {EventType}; {EventType} {Date}; {Date} {EventType}; {EventType} on {Date}; on {Date} {EventType}; {EventType} after {Time}; after {Time} {EventType}; {EventType} after {Date}; after {Date} {EventType}; {EventType} before {Time}; before {Time} {EventType}; {EventType} before {Date}; before {Date} {EventType}; {SubjectType} {HaveType} {EventType} after {Date}; after {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} after {Date}; after {Date} there {BeType} {EventType}; remind me that {SubjectType} {HaveType} {EventType}; remind me that there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} at {Time} {Date}; at {Time} {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} at {Time} {Date}; at {Time} {Date} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} by {Date}; by {Date} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} by {Date}; by {Date} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} by {Time}; by {Time} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} by {Time}; by {Time} there {BeType} {EventType}; make a note that {SubjectType} {HaveType} {EventType}; make a note that there {BeType} {EventType}; create a note that {SubjectType} {HaveType} {EventType}; create a note that there {BeType} {EventType}; add to my calendar that {SubjectType} {HaveType} {EventType}; add to my calendar that there {BeType} {EventType}; there {BeType} something happening later; there {BeType} event occurring later; there {BeType} something going on later; there {BeType} something coming up; |
| CreateNote | My friend's name is Raj; I have four children; My aunt is 65 years old; My dog is named Buddy; save money for a trip to France; keep a pencil on you to make notes; turn off the lights after leaving a room to conserve energy; Matt is a plumber; |
| CreateRecurringActionEvent | {SubjectType} {AuxiliaryVerbType} do something {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} do something; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} on {Date} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} on {Date}; on {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} at {Time} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} at {Time}; at {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} on {Date}; {RecurringType} on {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; on {Date} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} at {Time}; {RecurringType} at {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; at {Time} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; set a reminder that {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; set a reminder that {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType}; {ActionEventType} by {Time} {RecurringType}; {ActionEventType} {RecurringType} by {Time}; by {Time} {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType} by {Time}; by {Time} {RecurringType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} before {Time} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} before {Time}; before {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} before {Time}; {RecurringType} before {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; before {Time} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {ActionEventType} on {Date} {RecurringType}; {ActionEventType} {RecurringType} on {Date}; on {Date} {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType} on {Date}; {RecurringType} on {Date} {ActionEventType}; on {Date} {RecurringType} {ActionEventType}; {ActionEventType} after {Time} {RecurringType}; {ActionEventType} {RecurringType} after {Time}; after {Time} {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType} after {Time}; {RecurringType} after {Time} {ActionEventType}; after {Time} {RecurringType} {ActionEventType}; {ActionEventType} after {Date} {RecurringType}; {ActionEventType} {RecurringType} after {Date}; after {Date} {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType} after {Date}; {RecurringType} after {Date} {ActionEventType}; after {Date} {RecurringType} {ActionEventType}; {ActionEventType} before {Time} {RecurringType}; {ActionEventType} {RecurringType} before {Time}; before {Time} {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType} before {Time}; {RecurringType} before {Time} {ActionEventType}; before {Time} {RecurringType} {ActionEventType}; {ActionEventType} before {Date} {RecurringType}; {ActionEventType} {RecurringType} before {Date}; before {Date} {ActionEventType} {RecurringType}; {RecurringType} {ActionEventType} before {Date}; {RecurringType} before {Date} {ActionEventType}; before {Date} {RecurringType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} after {Date} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} after {Date}; after {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} after {Date}; {RecurringType} after {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; after {Date} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; remind me to {ActionEventType} {RecurringType}; {RecurringType} remind me to {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} at {Time} {Date} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} at {Time} {Date}; at {Time} {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} at {Time} {Date}; {RecurringType} at {Time} {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; at {Time} {Date} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} by {Date} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} by {Date}; by {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} by {Date}; {RecurringType} by {Date} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; by {Date} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} by {Time} {RecurringType}; {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType} by {Time}; by {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType} by {Time}; {RecurringType} by {Time} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; by {Time} {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; make a note that {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; make a note that {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; create a note that {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; create a note that {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; add to my calendar that {SubjectType} {AuxiliaryVerbType} {ActionEventType} {RecurringType}; add to my calendar that {RecurringType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; |
| CreateRecurringEvent | {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} {RecurringType}; {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} on {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} on {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} on {Date}; on {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} on {Date} {SubjectType} {HaveType} {EventType}; on {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} on {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} on {Date}; {RecurringType} there {BeType} {EventType} on {Date}; on {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} on {Date} there {BeType} {EventType}; on {Date} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} at {Time} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} at {Time}; {RecurringType} {SubjectType} {HaveType} {EventType} at {Time}; at {Time} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} at {Time} {SubjectType} {HaveType} {EventType}; at {Time} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} at {Time} {RecurringType}; there {BeType} {EventType} {RecurringType} at {Time}; {RecurringType} there {BeType} {EventType} at {Time}; at {Time} there {BeType} {EventType} {RecurringType}; {RecurringType} at {Time} there {BeType} {EventType}; at {Time} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} {Date}; {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} {Date} {SubjectType} {HaveType} {EventType}; {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} {Date}; {RecurringType} there {BeType} {EventType} {Date}; {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} {Date} there {BeType} {EventType}; {Date} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} {Date} {Time} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} {Date} {Time}; {RecurringType} {SubjectType} {HaveType} {EventType} {Date} {Time}; {Date} {Time} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} {Date} {Time} {SubjectType} {HaveType} {EventType}; {Date} {Time} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} {Date} {Time} {RecurringType}; there {BeType} {EventType} {RecurringType} {Date} {Time}; {RecurringType} there {BeType} {EventType} {Date} {Time}; {Date} {Time} there {BeType} {EventType} {RecurringType}; {RecurringType} {Date} {Time} there {BeType} {EventType}; {Date} {Time} {RecurringType} there {BeType} {EventType}; set a reminder that {SubjectType} {HaveType} {EventType} {RecurringType}; set a reminder that {RecurringType} {SubjectType} {HaveType} {EventType}; set a reminder that there {BeType} {EventType} {RecurringType}; set a reminder that {RecurringType} there {BeType} {EventType}; {EventType} {RecurringType}; {RecurringType} {EventType}; {EventType} by {Time} {RecurringType}; {EventType} {RecurringType} by {Time}; {RecurringType} {EventType} by {Time}; by {Time} {EventType} {RecurringType}; {RecurringType} by {Time} {EventType}; by {Time} {RecurringType} {EventType}; {SubjectType} {HaveType} {EventType} before {Time} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} before {Time}; {RecurringType} {SubjectType} {HaveType} {EventType} before {Time}; before {Time} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} before {Time} {SubjectType} {HaveType} {EventType}; before {Time} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} before {Time} {RecurringType}; there {BeType} {EventType} {RecurringType} before {Time}; {RecurringType} there {BeType} {EventType} before {Time}; before {Time} there {BeType} {EventType} {RecurringType}; {RecurringType} before {Time} there {BeType} {EventType}; before {Time} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} sometime {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} sometime {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} sometime {Date}; sometime {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} sometime {Date} {SubjectType} {HaveType} {EventType}; sometime {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} sometime {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} sometime {Date}; {RecurringType} there {BeType} {EventType} sometime {Date}; sometime {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} sometime {Date} there {BeType} {EventType}; sometime {Date} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} later {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} later {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} later {Date}; later {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} later {Date} {SubjectType} {HaveType} {EventType}; later {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} later {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} later {Date}; {RecurringType} there {BeType} {EventType} later {Date}; later {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} later {Date} there {BeType} {EventType}; later {Date} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} earlier {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} earlier {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} earlier {Date}; earlier {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} earlier {Date} {SubjectType} {HaveType} {EventType}; earlier {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} earlier {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} earlier {Date}; {RecurringType} there {BeType} {EventType} earlier {Date}; earlier {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} earlier {Date} there {BeType} {EventType}; earlier {Date} {RecurringType} there {BeType} {EventType}; {EventType} {Date} {RecurringType}; {EventType} {RecurringType} {Date}; {RecurringType} {EventType} {Date}; {Date} {EventType} {RecurringType}; {RecurringType} {Date} {EventType}; {Date} {RecurringType} {EventType}; {EventType} on {Date} {RecurringType}; {EventType} on {RecurringType} {Date}; {RecurringType} {EventType} on {Date}; on {Date} {EventType} {RecurringType}; {RecurringType} on {Date} {EventType}; on {Date} {RecurringType} {EventType}; {EventType} after {Time} {RecurringType}; {EventType} {RecurringType} after {Time}; {RecurringType} {EventType} after {Time}; after {Time} {EventType} {RecurringType}; {RecurringType} after {Time} {EventType}; after {Time} {RecurringType} {EventType}; {EventType} after {Date} {RecurringType}; {EventType} {RecurringType} after {Date}; {RecurringType} {EventType} after {Date}; after {Date} {EventType} {RecurringType}; {RecurringType} after {Date} {EventType}; after {Date} {RecurringType} {EventType}; {EventType} before {Time} {RecurringType}; {EventType} {RecurringType} before {Time}; {RecurringType} {EventType} before {Time}; before {Time} {EventType} {RecurringType}; {RecurringType} before {Time} {EventType}; before {Time} {RecurringType} {EventType}; {EventType} before {Date} {RecurringType}; {EventType} {RecurringType} before {Date}; {RecurringType} {EventType} before {Date}; before {Date} {EventType} {RecurringType}; {RecurringType} before {Date} {EventType}; before {Date} {RecurringType} {EventType}; {SubjectType} {HaveType} {EventType} after {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} after {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} after {Date}; after {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} after {Date} {SubjectType} {HaveType} {EventType}; after {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} after {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} after {Date}; {RecurringType} there {BeType} {EventType} after {Date}; after {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} after {Date} there {BeType} {EventType}; after {Date} {RecurringType} there {BeType} {EventType}; remind me that {SubjectType} {HaveType} {EventType} {RecurringType}; remind me that {RecurringType} {SubjectType} {HaveType} {EventType}; remind me that there {BeType} {EventType} {RecurringType}; remind me that {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} at {Time} {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} at {Time} {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} at {Time} {Date}; at {Time} {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} at {Time} {Date} {SubjectType} {HaveType} {EventType}; at {Time} {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} at {Time} {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} at {Time} {Date}; {RecurringType} there {BeType} {EventType} at {Time} {Date}; at {Time} {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} at {Time} {Date} there {BeType} {EventType}; at {Time} {Date} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} by {Date} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} by {Date}; {RecurringType} {SubjectType} {HaveType} {EventType} by {Date}; by {Date} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} by {Date} {SubjectType} {HaveType} {EventType}; by {Date} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} by {Date} {RecurringType}; there {BeType} {EventType} {RecurringType} by {Date}; {RecurringType} there {BeType} {EventType} by {Date}; by {Date} there {BeType} {EventType} {RecurringType}; {RecurringType} by {Date} there {BeType} {EventType}; by {Date} {RecurringType} there {BeType} {EventType}; {SubjectType} {HaveType} {EventType} by {Time} {RecurringType}; {SubjectType} {HaveType} {EventType} {RecurringType} by {Time}; {RecurringType} {SubjectType} {HaveType} {EventType} by {Time}; by {Time} {SubjectType} {HaveType} {EventType} {RecurringType}; {RecurringType} by {Time} {SubjectType} {HaveType} {EventType}; by {Time} {RecurringType} {SubjectType} {HaveType} {EventType}; there {BeType} {EventType} by {Time} {RecurringType}; there {BeType} {EventType} {RecurringType} by {Time}; {RecurringType} there {BeType} {EventType} by {Time}; by {Time} there {BeType} {EventType} {RecurringType}; {RecurringType} by {Time} there {BeType} {EventType}; by {Time} {RecurringType} there {BeType} {EventType}; make a note that {SubjectType} {HaveType} {EventType} {RecurringType}; make a note that {RecurringType} {SubjectType} {HaveType} {EventType}; make a note that there {BeType} {EventType} {RecurringType}; make a note that {RecurringType} there {BeType} {EventType}; create a note that {SubjectType} {HaveType} {EventType} {RecurringType}; create a note that {RecurringType} {SubjectType} {HaveType} {EventType}; create a note that there {BeType} {EventType} {RecurringType}; create a note that {RecurringType} there {BeType} {EventType}; add to my calendar that {SubjectType} {HaveType} {EventType} {RecurringType}; add to my calendar that {RecurringType} {SubjectType} {HaveType} {EventType}; add to my calendar that there {BeType} {EventType} {RecurringType}; add to my calendar that {RecurringType} there {BeType} {EventType}; there {BeType} something happening {RecurringType}; {RecurringType} there {BeType} something happening; there {BeType} event occurring {RecurringType}; {RecurringType} there {BeType} event occurring; there {BeType} something going on {RecurringType}; {RecurringType} there {BeType} something going on; there {BeType} something coming up {RecurringType}; {RecurringType} there {BeType} something coming up; |
| FallbackIntent | \*\*\*Closing Response: I'm sorry, but I don't understand. |
| Goodbye | bye; later; see you later; goodbye; see ya; have a nice day; good night; I have to go; catch you later; I got to go; alright I got things to do; talk to you later; |
| Hello | hi; hello; hey; hi there; greetings; howdy; good day; nice to meet you; |
| HowAreYou | how are you; how is it going; how are you doing; what is up; how have you been; you all right; what is new with you; how is it hanging; what is going on with you; how is everything; |
| Insult | you are stupid; you sound dumb; you are annoying; I hate you; I do not like you; shut up; xxxx you; you suck; you are useless; eat xxxx; dumbass; xxxx off; |
| LastThingSaid | what was I saying; what was I talking about; what did I just say; what am I talking about; I lost my train of thought; I forgot what I was saying; I forgot what I was talking about; I do not know what am I talking about; what was the last thing I said; |
| SearchNotes | {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {QuestionAuxiliaryVerbType} {ActionEventType}; {QuestionType} {QuestionAuxiliaryVerbType} {ActionEventType}; is there something {SubjectType} {AuxiliaryVerbType} do; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} do later; on what date {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; at what time {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} {ActionEventType}; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} doing later; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} do; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} get; {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} do something; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} {AuxiliaryVerbType} do something; {QuestionAuxiliaryVerbType} {SubjectType} {ActionEventType}; {QuestionType} {QuestionAuxiliaryVerbType} {SubjectType} {ActionEventType}; {QuestionAuxiliaryVerbType} {SubjectType} doing something later; |
| ThankYou | thanks; thank you; thanks a lot; thank you very much; I appreciate it; I appreciate your help; I just wanted to say thanks; I thank you for your time; it means a lot to me; I would be lost without you; thank you for helping me; I am grateful; |
| UserLocation | where am I; what is this place; where the xxxx am I; I am lost; I do not know where I am; I'm lost; |
| WhatIsYourName | what is your name; what are you called; what do you go by; do you have a name; who are you; |
| WhyAmIHere | where am I going; I have no idea where I am going; where am I heading; why am I here; I do not know where I am walking; I do not know why I am driving; what is my destination; |

Finally, within the Amazon Web Service Lex console, you must first add in the different **slot prompts, messages, types,** and **closing responses**. Note that some of the **closing responses** are messages sent back to the smartphone app. Below, you will find a Table 9 with the slot prompts and closing response values:

**Table 9**

*Amazon Lex Slot Types and Closing Responses Configuration*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Description** | **Slots** | | | | | **Closing Responses** |
| **Prompt for slot** | **Message** | | **Slot Type** | |  |
| AppHelpCalendar |  | | | | | calendar |
| AppHelpChecklist | checklist |
| AppHelpCloud | cloud |
| AppHelpFont | font |
| AppHelpLanguage | language |
| AppHelpMic | mic |
| AppHelpNotes | notes |
| AppHelpNotifications | notifications |
| AppHelpSecurity | security |
| AppHelpSound | sound |
| AppHelpTheme | theme |
| AppHelpTrigger | trigger |
| AppNavCalendar | calendar |
| AppNavChecklist | checklist |
| AppNavCloud | cloud |
| AppNavHelp | help |
| AppNavMenu | menu |
| AppNavMic | mic |
| AppNavNotes | notes |
| AppNavNotifications | notifications |
| AppNavSecurity | security |
| AppNavSettings | settings |
| AppNavTrigger | trigger |
| Compliment | Thanks. |
| CreateActionEvent | ActionEventType | | Do what? | | ActionEventType |  |
| SubjectType | | Who will? | | SubjectType |
| AuxiliaryVerbType | | What is the aux verb? | | ActionEventType |
| Date | | On what date? | | AMAZON.Date |
| Time | | At what time? | | AMAZON.Time |
| CreateEvent | EventType | | What is the event? | | EventType |
| SubjectType | | Who? | | SubjectType |
| HaveType | | Has | | HaveType |
| BeType | | Be | | BeType |
| Date | | On what date? | | AMAZON.Time |
| Time | | At what time? | | AMAZON.Time |
| CreateNote |  | | | | | [CREATE NOTE] I have created a new note for you. |
| CreateRecurringActionEvent | ActionEventType | | Do what? | | ActionEventType |  |
| SubjectType | | Who will? | | SubjectType |
| AuxiliaryVerbType | | What is the aux verb? | | ActionEventType |
| Date | | On what date? | | AMAZON.Date |
| Time | | At what time? | | AMAZON.Time |
| RecurringType | | How often? | | RecurringType |
| CreateRecurringEvent | EventType | | What is the event? | | EventType |
| SubjectType | | Who? | | SubjectType |
| HaveType | | Has | | HaveType |
| BeType | | Be | | BeType |
| Date | | On what date? | | AMAZON.Time |
| Time | | At what time? | | AMAZON.Time |
| RecurringType | | How often? | | RecurringType |
| FallbackIntent |  | | | | | I'm sorry, but I don't understand. |
| Goodbye | Goodbye. |
| Hello | Hello. |
| HowAreYou | I'm good. Thanks for asking. |
| Insult | I'm sorry you feel that way. |
| LastThingSaid |  |
| SearchNotes | QuestionAuxiliaryVerbType | | Who is doing what | | QuestionAuxiliaryVerbType |  |
| QuestionType | | What is the focus of the question? | | QuestionType |
| ActionEventType | | Do what? | | ActionEventType |
| SubjectType | | Who will? | | SubjectType |
| AuxiliaryVerbType | | What is the aux verb? | | ActionEventType |
| ThankYou |  | | | | | You're welcome. |
| UserLocation |  |
| WhatIsYourName | I'm Sam. |
| WhyAmIHere | WhyAmIHere |

# Testing the NLU Module

The NLU module will be tested by ensuring the correct inputs from the app process and send the correct outputs back to the app.

## Testing Objectives

### Unit Tests

A unit test assesses a single function, method, or class. The goal is to verify the accuracy of a logical unit.

**Table 10**

NLU Module Unit Tests

|  |  |  |
| --- | --- | --- |
| General Tests: Application Help | iPhone | Android |
| 1. “How do I make a note”  Success case: Display “Let me help you with that.” in the chat window. |  |  |
| 2. “How do I change choose a different language”  Success case: Direct the user to the help screen for the app feature |  |  |
| App Navigation | | |
| 3. “Take me to the main menu”  Success case: Display “Taking you to the Menu screen.” in the chat window. |  |  |
| 4. “I want to see my notes”  Success case: Direct the user to the screen identified from the text. |  |  |
| Answering | | |
| 10. “How’s it going?”  Success case: Take to the appropriate screen and display it in the chat window. |  |  |
| Creating a note | | |
| 15. “I need to pick up eggs tomorrow”  Success case: Create a new note with what is given.  Display “I created a note for you.” in the chat window.  (Optional) Direct the user to the notes screen. |  |  |
| User Location | | |
| 19. “Where am I?”  Success case: Determine the user’s location via an API or library  Either display their location (e.g., open Google Maps) or state their location (e.g., “You are at Park Place in Townsville, CA.”). |  |  |

### Integration Tests

An integration test assesses the application as a whole or a large part of it. The goal is to verify that all features tested can work together as expected.

See Table 6.

## Testing Procedures

1. TODO
2. Execute the application
3. Manually test each feature of the application
4. Document the test results
5. If any of the required tests fail, fix the issues, and collaborate with NLU team, DevOps and smartphone app team members
6. Rerun the updated failed tests
7. Complete the test results document
8. Complete the test report documentation

# Appendices

## Credits and Contributions

Below are the members that contributed or will contribute to the development of this application in alphabetical order:

|  |
| --- |
| · Addisu Worku |
| · Alec Baileys |
| · Andreas Cabrales |
| · Andrew Rohn |
| · Austin Johnson |
| · Ayodeji Famudehin |
| · Benjamin Cushing |
| · Chauntika Broggin |
| · Christian Ahmed |
| · Christian Cruz Jimenez |
| · Damion Sevilla |
| · Daniel Avery |
| · Debashis Jena |
| · Didimus Kimbi |
| · Elshaday Mesfin |
| · Endalkachew Girma |
| · Eskedar Endashw |
| · Firehiwot Chari |
| · Jeroen Soeurt |
| · Johnny Lockhart |
| · Joseph Kalfus |
| · Karen Crumb |
| · Kevin Bell |
| · Leela Subramanian |
| · Maddison Dunning |
| · Malik Webster |
| · Matthew Setiawan |
| · Michael Le |
| · Michelle Monfort |
| · Mitchell Olshansky |
| · Mod Drammeh |
| · Nicholas Ballo |
| · Obinna Okonkwo |
| · Presley Muwan |
| · Prince Antwi Aboagye |
| · Raul Benavides |
| · Rebecca Johnson |
| · Robert Wilson |
| · Roy Gordon |
| · Sami Salim |
| · Shawn Kelly |
| · Teresa Balbi |
| · Tyler Puschinsky  · Dr. Mir Mohammed Assadullah |