

System Design Document

Unit Converter Application

Maria Harrson

CS 6387: Topics in Software Engineering

Vanderbilt University

Nashville, TN

maria.harrison@vanderbilt.edu

1. Introduction

1.1. Overview

CupChameleon is a mobile application designed to offer a solution to users who need to convert liquid and dry units of measurement. By addressing the need for accurate and efficient measurement conversions while in the kitchen, this app may serve both casual users and professionals who require precise data for cooking, baking, and various scientific applications. The app supports conversions between metric and imperial units, ensuring flexibility for users across different regions and industries.

1.2. Purpose

The purpose of this system design document is to compile all of the information necessary to understand and implement the functionality of the CupChameleon application; an application that provides unit measurement conversions for cooking ingredients. This document will establish the CupChameleon applications purpose; potential applications; constraints; functional and non-functional requirements; hardware and software interfaces; design requirements; and potential use cases.

This document is written for two main audiences: the stakeholder(s) and the individual responsible for developing and deploying the project. To ensure that all parties involved with the project share the same level of understanding, definitions of all acronyms, abbreviations, and technical jargon used in this system design document will be provided. This document will need to be approved by the stakeholder before it is finalized to ensure that the stakeholder's expectations are properly accounted for before formal development on the system begins.

1.3. Scope

The unit converter system accepts numeric data representing the current measurement, to a single predefined unit of measurement. Additionally the system accepts encoded data from user selection input describing the unit of measurement for conversion. All input data provided by the user must align with the list of predefined units of measurements provided by the stakeholder. Immediately after the user inputs the data, it is processed through a conversion algorithm and converts the value for all units from the list of predefined units of measurements. Only one value is returned to the user, based on their initially requested unit of measurement. The CupChameleon system supports both metric and imperial unit systems.

A key objective in this project is to perform unit conversion with a precision up to the thousandths decimal point.

2. General Description

2.1. Product Functions

This application accepts a numeric value of the input measurement, the input unit value, the target unit value, and the unit system. Then, the system will calculate and convert the input value to all available units of measurement for the selected unit system. The conversion values are stored in a local cache. Then the unit conversion value is returned to the user.

The application will first require the user to select their preferred measurement system via a toggle button. They will select either Metric or Imperial. This system value is stored in the application's cache.

Next the user will enter the input measurement value; a numeric value of the measurement and its unit. The user will type in a text input field using their mobile device's keyboard. This value is stored. Then the user will select the input unit from a dropdown field. This value is stored. The dropdown field value options are limited to a predefined list of units for the selected measurement system.

Now that the input value and unit is stored, the user will now select the target unit. The user will select the target unit from a dropdownfield. This value is stored. The dropdown field value options are limited to a predefined list of units for the selected measurement system.

The user will click on the "Convert" button to start the conversion. The stored input data is passed to the conversion algorithm. The algorithm will perform a conversion calculation based on input unit and target unit. Once the calculation is complete, the value is stored and returned to the user.

2.2. User Characteristics

The end users of this product will be consumers who have a smart mobile phone. Additionally, those who need quick conversions for a liquid and dry measurement. Potential uses could include cooking measurement and scientific measurement conversions. Due to the prevalence of mobile applications, the user is not expected to have a deep understanding of technical components found in this product. Instead, they should have the competency to interact with such a device via a simple user interface that abstracts these technical components. Furthermore, the user should be able to type a numeric value into a text field box, make their selection from a drop down field, and click on a button, without accommodation.

For cooking measurement conversion, a user may need to convert a recipe ingredient measurement from cups to milliliters. The user enters the quantity in cups, selects "Cups" as the input unit, and selects "Milliliters" as the target unit. The app performs the conversion and displays the result in milliliters. For scientific measurement conversion, a user may need to convert a volume measurement from liters to gallons. The user enters the quantity in liters, selects "Liters" as the input unit, and selects "Gallons" as the target unit. The app provides the converted volume in gallons promptly.

2.3. Assumptions and Dependencies

It is assumed that only one user will utilize the system concurrently. This will prevent control flow collisions that would be inevitable in a system not intended to serve multiple users at once. The app maintains a list of predefined units as specified by stakeholders, ensuring that all conversions align with accepted standards. It supports both metric and imperial systems, allowing users to work with units relevant to their needs. The types of unit conversion available are constrained to the predefined unit list.

3. System Structure

3.1. Landing Page

The Landing Page is the first screen in the application. It is automatically displayed to the user every time they launch the app. This page is designed to be minimalistic with three core functionalities, displaying the title of the application, allowing the user to select their desired measurement system, and transitioning the user to the Inputs Page.

3.2. Measurement System Selection Toggle

While on the Landing Page, the user is prompted to select their preferred measurement system via a toggle button. The selection options are based on the predefined systems as specified by the stakeholders: Metric or Imperial. The system saves the user's selected measurement in a local cache and on subsequent app launches the measurement system is automatically set based on the last selection the user made/last stored value. The user is allowed to update or change their measurement system selection at any time.

3.3. Inputs Page

The Inputs Page is the second screen that is displayed to the user, and is only displayed after the Landing Page has displayed. This page is composed of all input fields the user interacts with, including: the input value textfield, the input unit dropdown selection field, the target unit

dropdown selection field, and the conversion button. This page's functionality is to accept user data input, start the conversion algorithm, and transition the user to the Results Page.

3.4. Input Value Text Field, Input Unit Dropdown & Target Unit Dropdown Selection

The Input Value Text Field, Input Unit Dropdown Section and Target Unit Dropdown Section are housed within the Inputs Page. The Input Value text field requires the user to enter a decimal value up to the hundredths decimal point via their keyboard. The Input Unit and Target Unit Dropdown Selection requires the user to select the desired input unit and target unit via a dropdown field. All three of these input fields must be selected/filled out for the conversion button to become enabled. As validation, if one of these fields is null or blank, the conversion button will be in a disabled state, and the user will be unable to proceed with the process.

3.5. Conversion Button

The Conversion Button is housed within the Inputs Page. Clicking this button will first call a script that stores all of the input data into the local cache and data model, then triggers the conversion algorithm to run. This button has two states, disabled and enabled, and by default is set to disabled. If the user does not fill out all input/dropdown fields on the Input Page, the button will remain in a disabled state. The button enters the enabled state when the user has all input fields filled out. Additionally, this button will also transition the user from the Input Page to the Results page.

3.6. Results Page

The Results Page is the third screen displayed to the user, and is only displayed after the Input Page when the user clicks the Convert button. This screen's purpose is to display a loading indicator when the algorithm is still running, and to provide feedback to the user and return the converted value.

3.7. Data Model

3.8. Conversion Algorithm

4. Specific Requirements

4.1. Business Requirements

BR. 1.0 The system must have a landing page that displays when the user launches the app.

- BR. 2.0** The system must have an inputs page that allows the user to enter the value and unit of the input measurement and the target unit.
- BR. 3.0** The system must have a text field and dropdown field on the inputs page that allows the user to enter the input measurement and select the input unit.
- BR. 4.0** The system must have a dropdown field on the inputs page that allows the user to select the target unit.
- BR. 5.0** The system must have a conversion button on the inputs page that allows the user to initialize the conversion process.
- BR. 6.0** The system must have a results page that returns and displays the converted measurement.
- BR. 7.0** The system must have a data model to store the user's input data.
- BR. 8.0** The system must have a conversion algorithm that performs arithmetic calculations.

4.2. Functional Requirements

- FR. 1.0** The system's landing page must display when the user launches the app.
- FR. 2.0** The landing page must require the user to select either metric or imperial measurement system via a toggle.
- FR. 3.0** The landing page must transition the user to the input page.
- FR. 4.0** The input page must display after the landing page. Additionally the inputs page must allow the user to input measurement values
- FR. 5.0** The input page must have a numeric input value text field. This text field must accept a decimal value up to and no more than the thousandths place.
- FR. 6.0** The input page must have a drop down list of predefined input units for the user to select from.
- FR. 7.0** The input page must have a drop down list of predefined output units for the user to select from.
- FR. 8.0** The input page must have a conversion button that is clickable.
- FR. 9.0** Clicking the conversion button triggers the conversion process. The conversion button will be attached to a script that calls the conversion algorithm.
- FR. 10.0** The conversion button must be disabled if the user does not fill out all the input fields.
- FR. 11.0** The conversion button must be enabled if the user has filled out all the input fields.
- FR. 12.0** The conversion button must send the user to the result page.
- FR. 13.0** The result page must display after the input page.
- FR. 14.0** The conversion button must send the user to the result page.
- FR. 15.0** The result page must display a loading indicator when the conversion algorithm is still running to notify the user that the converted value hasn't been returned yet.
- FR. 16.0** The result page must display the input measurement at the top of the screen. It must display and return the newly converted value on a text line below the input measurement.

FR. 17.0 The result page must display the input measurement at the top of the screen. It must display and return the newly converted value on a text line below the input measurement.

5. Change Management Process

Throughout the development of the project, software requirements may change based on the needs of the stakeholders and the progress the developers make. The stakeholder's needs may change during the development lifecycle. Changes from stakeholders often involve the functional requirements, and the developers' progress may require changes to be made to software requirements too. As the software is further developed and if time allows, additional non-functional requirements may be added to the program. The current scope of the project is to complete all functional requirements before progress is made on nonfunctional requirements. Developers' progress changes can also go in the opposite direction. In the case that certain requirements prove too challenging they may need to be redefined or removed. The process for any of these cases should be largely the same.

5.1. Operational Considerations

The mobile application will receive regular updates via the Google Play Store or Apple App Store to maintain the application and ensure the conversion algorithm remains accurate and up to date. Help and support features will be available in the application to assist users with any questions or issues regarding the app's functionality.