# Systems Engineering Management Plan

## Unit Converter Application

Maria Harrson
CS 6387: Topics in Software Engineering
Vanderbilt University
Nashville, TN
maria.harrison@vanderbilt.edu

| **Section** | **Content** |
|---|---|
| Title Page | SYSTEMS ENGINEERING MANAGEMENT PLAN FOR<br>CupChameleon- Unit Converter Application |
| 1.0 Purpose of Document | The purpose of this 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 CupChamelon applications purpose; potential applications; constraints; functional and non-functional requirements; hardware and software interfaces; design requirements; and potential use cases.<br><br>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. |
| 2.0 Overview, Purpose, and Scope of Project | **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.<br>**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.<br>**Objective:** A key objective in this project is to perform unit conversion with a precision up to the thousandths decimal point. |
| 3.0 Technical Summary | 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.<br>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. |
| 3.1 System Structure | 1. Input Data<br>The user enters the numeric value in the text field box and selects the unit of measurement for the input data from the predefined list of units and selects the measurement system for both the input and output data. The user selects their desired output unit of measurement for the converted value from a predefined list. |

2. Decision Gates/Data Validation

The calculate button will automatically be disabled if any of the following input data is null or empty:

- Selected Unit of Measurement for the conversion input
- Numeric text input field
- Selected System Unit of Measurement
- Selected Unit of Measurement for the conversion output

3. Storing the Input Data

The system stores all user input into its local storage. Each input data has its own storage. The stored values are passed into the conversion algorithm as parameters.

4. Conversion Algorithm

The app converts the value for the predefined unit in the selected measurement system and stores the calculated value in a local cache. Only one converted value is returned to the user via the interface at a time.

5. Output Results

Upon submitting the input data, the system processes the input data into three data types. The algorithm performs the conversion based on a predefined list of units and target output unit.

6. User Feedback

The user receives immediate feedback with the converted value displayed on the screen. Users can input new data or select

| | |
|---|---|
| | different units for further conversions as needed.<br>7. System Functionality<br>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. |
| 3.2 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.<br>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.<br>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.<br><br>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. |
| 4.0 Systems Engineering Process | **Functional Requirements:**<br>FR. 1 The UC system shall allow the user to type a numeric value in a text field.<br>FR. 2 The UC system shall allow the user to select their preferred system of measurement<br>FR. 3 The UC system shall allow the user to select an input unit from a dropdown field input.<br>FR. 4  The UC system shall allow the user to select a target unit from a dropdown field input.<br>FR. 5  The UC system shall allow the user to trigger the conversion when clicking a button<br>FR. 6 The UC system shall calculate a unit conversion using the input data from the user<br>FR. 7 The UC system shall provide the user with immediate feedback when the conversion algorithm is complete. It will return and display the converted value (in the unit selected by the user)<br>FR. 8 The UC system shall calculate unit conversions with a precision to the thousandths decimal place. |

| | |
|---|---|
| 5.0 Transitioning Critical Technologies | |
| 6.0 Integration of the System | |
| 7.0 Integration of the Systems Engineering Effort | The conversion algorithm will be written in Python. The UI elements including all input fields and buttons will be created with Material UI Kit. |
| 8.0 Operations and Maintenance Plan | 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. |