**MECHTRON 4TB6A / SFWRENG 4G06A**

**Deliverable 2 - Software Requirements Specification**

**Billsplitter Application**

**Fall/Winter 2019-2020**

**Group #26**

George Mo - 400028860

Aidan Schonewille - 400030270

Emilio Hajj - 001402245

Hamid Ghasemi - 400028420

Ibrahim Malik - 001049305

Prithvi Jethwa- 400050638

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Author | Description |
| 0.1 | 10/10/19 | Ibrahim Malik | Generated document template |
| 0.2 | 15/10/19 | Ibrahim Malik | Added content to general sections |
| 0.3 | 22/10/19 | Prithvi | Revisions to sections and organization |
| 1.0 | 23/10/19 | Emilio Hajj | Finalized sections 5,6,10 |
| 1.1 | 24/10/19 | Ibrahim Malik | Finialized sections 1-4 |
| 1.2 | 26/10/19 | Hamid Ghasemi | Completed specific requirements Section |
| 1.3 | 26/10/19 | George Mo | Finialized sections 11-12, 18-19 |
| 1.4 | 26/10/19 | Prithvi Jethwa | Completed Section 5-8 |
| 1.5 | 27/10/19 | Aidan Schonewille | Completed functional requirements section |
| 1.6 | 28/10/19 | Ibrahim Malik | Finalized requirements sections |

# 

# Table of Contents

[**Table of Contents**](#_b6v7rq55m3ca) **3**

[**Introduction**](#_xvvrhdpjfdkf) **4**

[Document Purpose](#_bx8qufwbiiut) 4

[System Scope](#_ht4bb34mcixp) 4

[2.1 Purpose](#_mvdixrz1besr) 4

[2.2 Goals](#_9ohv2b4iedxc) 4

[2.3 Project Scope](#_g8974pbb6fdm) 5

[Document Overview](#_8eyuex6aqey6) 5

[**General Description**](#_4z2e7w55l4sy) **6**

[Behavior Overview (System Overview)](#_ebua144bbdsz) 6

[Required behavior description](#_w4dw34xm3avr) 7

[User characteristics](#_n0e0wqqqufi) 9

[General constraints](#_e1u6g2byki89) 9

[Context Diagram](#_x8crmrm2q82s) 9

[Functional decomposition](#_sq8jrvdutgz6) 10

[**Requirements and Operation**](#_k1vpk46k9884) **11**

[Performance requirements](#_ijehkh4ogezx) 11

[Normal operation/Use cases](#_i55mrdhu8jf9) 11

[11.1 Application is initialized](#_swoxwl3h9w2n) 11

[11.2 User input](#_4uzr5sji8c27) 11

[11.3 Display of itemized bill](#_emyq9mdgt1da) 11

[12.4 User updates itemized bill](#_s0x3yumjbkt3) 11

[11.5 User downloads final itemized bill](#_wl8g2cumopt6) 12

[Undesired event handling](#_gyl4j7uhutkd) 12

[12.1 Incomplete/Partial Bill imaging](#_irvd2eu06nps) 12

[12.2 Unreadable Bill imaging](#_bnx9v2oykdal) 12

[**Specific Requirements**](#_w1zh335ki769) **12**

[List of requirements that are likely to change](#_abeyodigz4j) 12

[List of requirements that are not likely to change](#_n1qte6aqg3fr) 13

[Functional Requirements](#_62ws0fnt96b7) 13

[Non-Functional Requirements](#_helthofg19fm) 19

[**Appendix**](#_jmakmuhgpnft) **23**

[Monitored and controlled variables (with units)](#_j160p52gd8gb) 23

[Constants](#_chcu51k9wh6g) 23

[**References**](#_skcpv9xd4xtk) **24**

# **Introduction**

## Document Purpose

This document intends to provide a set of specifications and requirements that are necessary to verify the development of the Billsplitter mobile application. Outlined in this document is a general description of the application, users, inputs, constraints, functional decomposition of the system, functional requirements and nonfunctional requirements. It will ensure that the project’s development adheres to the intended objectives and captures the essential capstone requirements as well as providing a guideline for system design. The project is the creation of a mobile application that is capable of dividing a certain amount of money (derived from a bill or a receipt) into smaller amounts. It is further described in detail within the sections below.

## System Scope

#### 2.1 Purpose

The purpose of this capstone project is to develop and launch a mobile application that is capable of automating the tasks of splitting bills amongst multiple individuals. It will intend to serve as a final year capstone project for 6 final year McMaster Engineering Students that are enrolled in the Faculty of Computer and Software.

The application itself will allow users to divide a monetary amount that contains a sum of many items, into smaller amounts that correspond to the respective people who are individually responsible for those items. The application will be highly intuitive and simple to operate while generating an accurate output quickly. There will be 3 ways the user can enter the initial amount. Then the user will assign individual items to their respective owners, and the application will promptly display the amount owed by each of these owners. All of these amounts will sum up to the cumulative total displayed on the bill. This app will enable users to quickly request money from other members in their party and intends to significantly improve their experience of splitting their bills. By automating this process, there will no longer be any need to extensively analyse the bill manually or calculate debts.

#### 2.2 Goals

After consulting with Dr. Wassyng, the development team decided on a set of objectives will enable the final product to perform its duties of improving the user experience as intended. Outlined below are the minimum functionality requirements for the application:

1. The final product will be a mobile application that is available for download for any user on iOS and Google Play stores.
2. The user can manually enter in an amount to be split.
3. The final product will allow the user to take a picture of a bill/receipt or upload an old receipt
4. Image detection will allow the application identify key items of the receipt/bill
5. The user can assign multiple items to one user and assign multiple users to one item as well.
6. User can download itemized image of bill that will be stored in their device memory

Additionally, the purpose of the application is to serve as a capstone project that adheres to the following objectives:

1. A completed product satisfying all the functional and non-functional requirements is available by April 15, 2020
2. The development of the system does not cost more that $750 CAD

#### 2.3 Project Scope

This project enables the developmental team to apply the concepts they have learned within their academic careers and generate an application that can provide significant utility to users long after the completion of the capstone deadlines. The functionalities described above outline the desired objectives of the project.

## Document Overview

The document is organized categorically into 5 major sections. This section outlines the project objectives and purpose relative to the default client (Dr. Wassyng) and introduces the goals of the development team. The following section describes the general behaviour of the final product and an overarching high-level description of the system implementation. Section 3 and 4 pertain to the specific requirements and states demonstrated by the application. This will include functional and non-functional requirements along with any requirements that are scheduled to update or change. An appendix containing variable names, acronyms and abbreviations is included at the very end of the document along with a list of references.

This document intends to serve the development team, the teaching assistants in 4TB6 and Dr. Wassyng. Hence, it is tailored to deliver a mix of both high-level descriptions and technical details that define the project. The document is enumerated by subsections to make it easier to navigate.

# **General Description**

## Behavior Overview (System Overview)

The product at hand, known as the Billsplitter Application is a system that will be developed by six (6) final year McMaster Engineering Students that are enrolled in the Faculty of Computer and Software (CAS) for their Capstone Project. In the following text the system will also be known as the product or application by which the user will be handling. In the most abstract sense, the system must accept a photograph/image of a receipt as user input. The product must also accept generic user input instead of an image at its lowest level. The application must be able to itemize the content while allowing users to customize and assign payment methods. As output, the system must return the final cost owed by each individual.

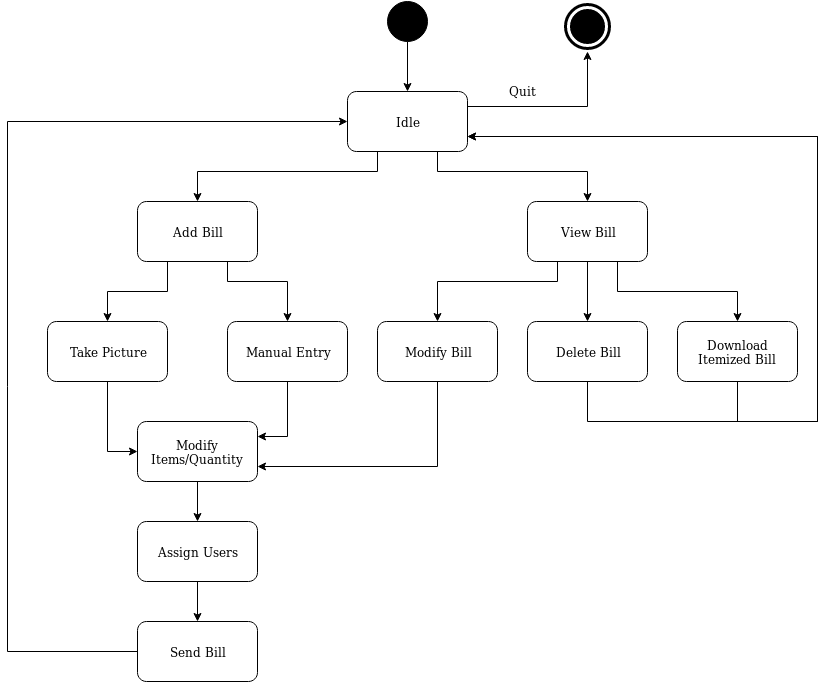
The two main functionalities that will encapsulate the entire project are as follows:

* The system will itemize and sort the contents of a bill or receipt
* The system will appropriately split bills accordingly.

Additionally, there will be an added functionality that can be incorporated into the project as follows:

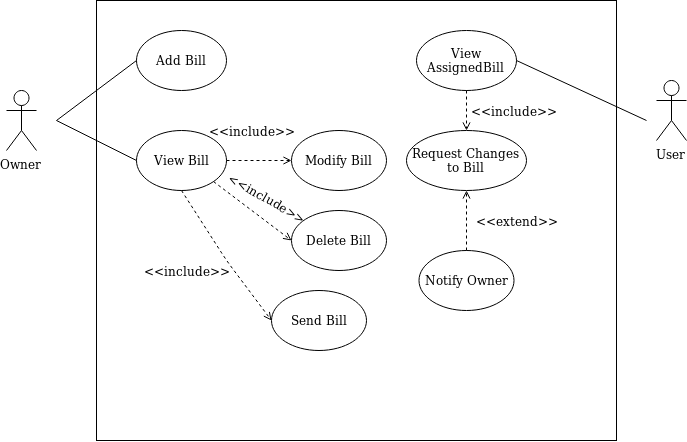
* The system will allow the user to download the itemized bill as a picture.

## Required behavior description



*Figure 1.0: State Machine Diagram*

|  |  |
| --- | --- |
| State | Description |
| Idle | In this state, the user will remain in the main menu of the application. |
| Add Bill | In this state, the user will add a new Receipt to be processed. |
| Take Picture | In this state, the user will choose the option to take a picture of the Receipt to be processed |
| Manual Entry (might not keep this) | In this state, the user will choose the option to manually add items to the virtual receipt. |
| Modify Items/Quantity | In this state, the user will modify the price and quantity for the items on the virtual receipt. |
| Assign Users | In this state, the user assigns items on the virtual receipt to the respective owners. |
| Send Bill | In this state, the user will send a notification to other users about the individual breakdown of the prices on the virtual receipt. |
| View Bill | In this state, the user will view past virtual receipts that have been settled or are waiting to be settled. |
| Modify Bill | In this state, the user will modify the individual virtual receipt that they own. |
| Delete Bill | In this state, the user will remove the receipt that they own. |
| Download Itemized Bill | In this state, the user will be able to download a new picture of the bill with items assigned to their respective owners |



*Figure 1.2: Use-Case Diagram. In this diagram, owner refers to the person uploading the image and using the application and the user refers to other individuals who are also going to be involved in splitting the bill and are recipients of individualized totals*

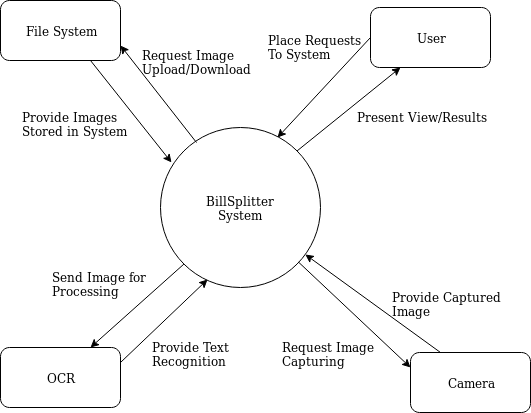
## User characteristics

The application is targeted to users that are between the age group of 17-70 years old. This includes users of varying skill levels from highschool students, University students or the general public. The application is developed purely for mobile users and hence the users are expected to own a smartphone that supports Android or iOS.

## General constraints

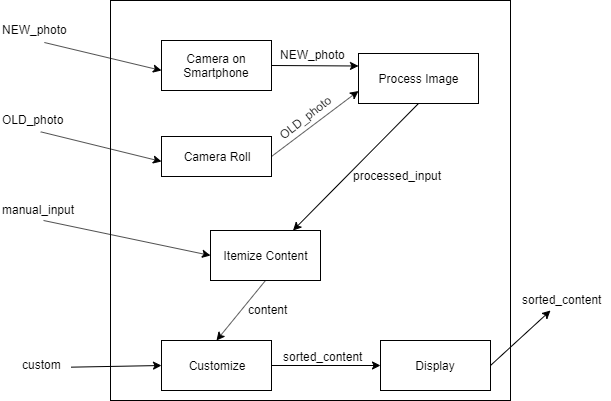
* Due to budget constraints, the application may suffer in performance when identifying items from bills using OCR. Vendors that provide OCR services may include fees upon using their services. Hence, usage of free OCR services will be used which may hinder the performance of the application.
* Due to time constraints, product development and balancing workload from university may affect the requirements that are implemented in the application

## Context Diagram



*Figure 1.3: Context Diagram*

## Functional decomposition

*Figure 1.4: Functional Decomposition Diagram*

The figure 1.4 depicts the functional decomposition of the system. The outer rectangular box represents the system. Similarly, the arrows coming into the system represent the user input variables and interactions from the environment and the arrow coming out of the system represent the system output variables to the environment.

Inside the rectangular box is the internal functionalities of the system in order to meet the client’s requirements. Each block has an assigned task designed to process, dissect and customize the data provided by the client’s input. The process image block, receives image files of a receipt and is responsible for breaking it down and extracting the necessary data. The itemize content block will organize the data in a manner by which the customize block can efficiently sort and calculate.

# **Requirements and Operation**

## Performance requirements

The application should be able to deliver accurate divisions of the bill to at least 95% of the time. This was an arbitrarily agreed upon number by the group members to validate the accuracy of splitting algorithm. It will be designed to be compatible with iOS and Android devices that run on at least the latest version of their respective operating system software. The objectives

## Normal operation/Use cases

#### 11.1 Application is initialized

Upon selecting the application icon on the user’s device, the program will be initialized and appear on the entire full screen displaying the main menu of the application. The user can select how they would like to enter their input.

#### 11.2 User input

The user can enter the input in 3 different ways once they are in the main screen. The user can upload an image from their device gallery, take a new image using their device camera app or manually enter type in the bill total. The user then has to specify the number of people who will be splitting the bill.

#### 11.3 Display of itemized bill

The system will detect all the items on the bill along with the corresponding numerical cost of each of these items. Thus the bill will be individualized depending on the contents and can operate individually. The user will be able to see this itemized bill on their phone screen itself.

#### 12.4 User updates itemized bill

The user can make modifications to the itemized bill. They can update item, if the price is incorrectly shown. The user can assign individuals to respective items and thus the price of that item will be added to that individual’s total contribution. By

#### 11.5 User downloads final itemized bill

The user can download an image of the itemized bill that will be stored in the device memory.

## Undesired event handling

#### 12.1 Incomplete/Partial Bill imaging

The system shall notify the user and prompt for a retake of the bill image

#### 12.2 Unreadable Bill imaging

The system shall notify the user and prompt for a retake of the bill image

# **Specific Requirements**

## List of requirements that are likely to change

1. Application shall store an image of the bill in small size in kb.

Rationale: This requirement could be changed based on the performance. If application takes a long time to compress the picture, it might just store it whatever the size is.

1. Application shall have colorful, intuitive background.

Rationale: This requirement is not necessary overall but it might make a difference in user experience.

1. Application will not distribute the image of the bill over the internet or use the internet to access external APIs to process the image in any step of the process.

Rationale: As it stands right now, it is not necessary to access an external sources to generate any output. The algorithm to split the bill is designed to be within the application itself. Hence there is no need to access data. However if during the development of the project, there is a need to access data/internet to enhance the application, then this requirement will change.

1. Application will allow user to download the itemized image

Rationale: Currently, this implementation of the application will allow the user to download the itemized image and save it in their device memory. However, there is potential of adding services to share this photo via text or email from within the application thereby increasing its utility and further enhancing the user experience. This will only be sought if it is deemed feasible to add this function within the project’s time and budget constraints and decided after the first iteration of the fully-functioning product.

## List of requirements that are not likely to change

1. The application shall run very smoothly.

Rationale: This requirement is a fundamental part of the application. Program has to run smoothly to provide a good service for clients.

1. Application shall operate image processing very fast.

Rationale: Image processing requirement causes app to run and execute faster.

1. Application shall enable users to modify the bill in a simple way.

Rationale: Modifying the bill in simple manner allows user to use the interface easily. This means users can change the amount of bills whenever bill is not recognized by the image processing.

1. Application shall have an efficient algorithm to split the bills.

Rationale: The whole purpose of this application is to split the bill correctly. This requirement is essential so it won’t be changed through development of the application.

## Functional Requirements

15.1

Requirement: The application shall run on phones running Android 7 and above, or iOS 11 and above.

Rational: The application needs to run on common mobile operating systems used today, and most mobile phones run on these minimum versions.

Fit Criterion: The application must operate functionally on all Android 7 and above operating systems, and all iOS 11 and above operating systems.

Dependencies: N/A

Performance Requirements: The application should be compatible with the operating systems specified above.

Normal Operation:

* The application launches and operates on the specified operating systems, specified above.

15.2

Requirement: The application should launch to a central screen, allowing the user to select either taking a photo or uploading one from their device storage, or enter the bill item entry field.

Rational: There needs to be a point of entry into the application’s main features upon launching it.

Fit Criterion: The application launches to a page that allows access to taking or uploading a photo, or entering bill items manually..

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* The application launches with a main screen, and can navigate to either method of photo selection or bill entry.

15.3

Requirement: The application shall allow the user to take and store photos.

Rational: This is one of the main forms of bill input that we have decided upon, and is integral for loading the user’s bill.

Fit Criterion: The application can successfully take a photo, and store it in its memory.

Dependencies: N/A

Performance Requirements:

* The application must interface with the user’s camera application on their mobile device.

Normal Operation:

* The application redirects to a camera view, and once the photo is taken it is loaded into the application.

15.4

Requirement: The application shall allow the user to load a photo from their device’s memory.

Rational: This is one of the main forms of bill input that we have decided upon, and is integral for loading the user’s bill.

Fit Criterion: The application can successfully load a photo from the user’s, and store it in its memory.

Dependencies: N/A

Performance Requirements:

* The application must interface with the user’s camera roll or storage application on their mobile device.

Normal Operation:

* The application redirects to a list of photos, and once the photo is selected it is loaded into the application.

15.5

Requirement: The application shall accept a list of items and their costs, entered manually by the user on their mobile device.

Rational: This is another method of inputting a bill to the application, should the user’s camera not function or if the user decides to input a simpler bill to split that does not carry several individual items.

Fit Criterion: The application can receive input through the mobile device in the form of a list of items in a bill.

Dependencies: N/A

Performance Requirements:

* The application must accept keyboard input from the mobile device.

Normal Operation:

* The user types in their items and prices, and the application stores it as an itemized bill.

15.6

Requirement: The application shall parse a loaded image of a bill and itemize it into a list of items and prices.

Rational: One of the main features of our application is reading an image and converting it into an itemized bill, so this requirement is necessary.

Fit Criterion: The application can parse an image of a bill into a list and store the list in the application’s memory.

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* The image is loaded into the application, and an itemized list of costs is produced and stored.

15.7

Requirement: The application shall allow for the user to edit the list of items and prices.

Rational: In case of an error in image parsing, there should be a fallback for the user to override the parsed results.

Fit Criterion: The application can take user input to modify the parsed list.

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* The application accepts input from the user to modify an entry, and it is successfully modified and the list updated.

15.8

Requirement: The application shall allow the user to assign different list items to different group members.

Rational: This is one of the key functionalities of our application, and is needed to split the bill between members.

Fit Criterion: The application can take user input to assign a list item to a person.

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* The application accepts input from the user to assign an entry, and it is successfully modified and the list updated.

15.9

Requirement: The application shall allow the user to assign multiple items to one individual.

Rational: This is one of the key functionalities of our application, and is needed to split the bill between members.

Fit Criterion: The application can take user input to assign several list items to a single person.

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* The application accepts input from the user to assign an entry, and it is successfully modified and the list updated.

15.10

Requirement: The application shall allow the user to multiple individuals to one list item.

Rational: This is one of the key functionalities of our application, and is needed to split the bill between members especially when multiple members utilize or share the same item.

Fit Criterion: The application can take user input to assign several individuals to a single item.

Dependencies: N/A

Performance Requirements:

* N/A

Normal Operation:

The application accepts input from the user to assign an entry, and it is successfully modified and the list updated.

15.11

Requirement: The application shall generate individual bills and totals (including appropriately proportional tax amount) for each member assigned on the list after items have been assigned.

Rational: The application needs to generate the “split” bills once items have been assigned to members in order to send bills to each member. The tax will also be added on top of individual summing amounts thereby allowing the proportional distribution of the total tax on the bill as well.

Fit Criterion: The application can take user input to assign a list item to a person.

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* The application accepts input from the user to assign an entry, and it is successfully modified and the list updated.

## Non-Functional Requirements

16.1

Requirement: The application shall run without errors and any occurrence of sudden visual glitches.

Rational: Having an application that run smooth provides better experience for the users.

Fit Criterion: It shall provide smooth touch screen access for the users since this is a phone application.

Dependencies: N/A

Performance Requirements:

* Users shall switch between each part of the application without any glitch.

Normal Operation:

* The program shall execute through each step very smoothly till termination of the program.

16.2

Requirement: The application shall operate image processing very fast.

Rational: Having an application that can complete the operation of image processing successfully and do it fast is major in this project.

Fit Criterion: It shall complete the task under 5 seconds.

Dependencies: N/A

Performance Requirements:

* Having an application which complete processing of one image under t\_exe.

Normal Operation:

* The program shall execute and image processing completes the task during the demanding period.

16.3

Requirement: Application shall enable users to modify the bill in a simple way.

Rational: Having an app which user can modify the bill easily provides a better experience for the user.

Fit Criterion: App shall provide a place for user to input the number for parts they like to get modified.

Dependencies: N/A

Performance Requirements:

* Having an application which modify the documents easily and conveniently for users.

Normal Operation:

* The app shall modify the bill whenever user’s request very easily and smoothly.

16.4

Requirement: The application shall store an image of the bill in small size in kb.

Rational: This requirement will reduce image taking up space on a phone.

Fit Criterion: It shall complete store an image less than 20kb.

Dependencies: N/A

Performance Requirements:

* Application will store an image with size smaller than s\_size.

Normal Operation:

* User asks to get an image of the bill and application stores it on the user’s phone with a size of less than 20kb

16.5

Requirement: The application shall have an efficient algorithm to split the bills.

Rational: With this requirement, application would get a result faster.

Fit Criterion: It shall complete the task under 5 seconds.

Dependencies: N/A

Performance Requirements:

* Having an application which complete the calculation under 5 seconds.

Normal Operation:

* The user demands to get a result and it would be provided to the user during the demanding period.

16.6

Requirement: Application shall have colorful background.

Rational: This requirement has psychological aspect.

Fit Criterion: It shall have a neat and colorful background.

Dependencies: N/A

Performance Requirements:

* Background is colorful until termination.

Normal Operation:

* After user clicks on application, application opens with a colorful background.

16.7

Requirement: Application will not distribute the image of the bill over the internet or use the internet to access external APIs to process the image in any step of the process.

Rational: To ensure the security of the data is adhered.

Fit Criterion: It shall ensure the security of data by not distributing on the internet.

Dependencies: N/A

Performance Requirements:

* App calculates and stores a picture on user’s phone by asking user to give it permission.

Normal Operation:

* After calculating and splitting the bill, app stores a picture on user’s phone and does not upload it on the internet.

16.8

Requirement: Application shall be easy to use for everyone age 17 above.

Rational: By making application convenient, user can get results quickly without any effort.

Fit Criterion: Application would be easy to use.

Dependencies: N/A

Performance Requirements:

* Apps opens, runs with user interaction, calculate and store an output on users phone.

Normal Operation:

* Users don’t need to spend a lot of time to figure out what they should do to get results.

16.9

Requirement: Application shall be in English and accept English bills only.

Rational: By making application English, we are accepting the most universally recognized language and targeting English-speaking users for the first iteration. This version of the application will not support multiple languages.

Fit Criterion: Application will be in English language only.

Dependencies: N/A

Performance Requirements: N/A

Normal Operation:

* Users will decipher instructions in English and interact with the application based on the comprehension of that language.

# **Appendix**

## Monitored and controlled variables (with units)

|  |  |  |
| --- | --- | --- |
| Variable Name | Units | Description |
| individual\_sum | $ | The updated cost of the bill to the individual pertaining to their items |
| num\_users | - | The total number of individuals amongst who the bill will be split |
| total\_sum | $ | Total final value of the bill containing all the individual items and the tax. |

## Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Value | Units | Description |
| s\_size | 5000 | kB | Maximum size at which processed images are saved to the device |
| t\_exe | 20 | s | Maximum time to process one (1) bill |

# 

# **References**

*This document was produced using a template derived from IEEE*

[1] *830-1998 - IEEE Recommended Practice for Software Requirements Specifications - IEEE Standard*, https://ieeexplore.ieee.org/document/720574/.