

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**SYSTEM REQUIREMENTS SPECIFICATION
CSE 4316: SENIOR DESIGN I
SUMMER 2021**



**CODING AVENGERS
WINE INVENTORY MANAGEMENT**

**TUYEN VO
SITA LAMA
MOKSHADA UPRETI
LINH TRAN
GREG WHATLEY**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	07.15.2021	GW	document creation
1.0	07.30.2021	TV, SL, MU, LT, GW	First draft

CONTENTS

1	Product Concept	6
1.1	Purpose and Use	6
1.2	Intended Audience	6
2	Product Description	7
2.1	Features & Functions	7
2.2	External Inputs & Outputs	7
2.3	Product Interfaces	7
3	Customer Requirements	8
3.1	Get User First Name	8
3.2	Get User Name	8
3.3	Get User Email	8
3.4	Get User Password	9
3.5	Create Account	9
3.6	Verify User Email	10
3.7	Log In User	10
3.8	Log Out User	10
3.9	Change User Password	11
3.10	Change User Name	11
3.11	Delete User	11
3.12	Get Bottle Brand	12
3.13	Get Wine Color	12
3.14	Get Wine Type	12
3.15	Get Bottle Status	13
3.16	Get Bottle Quantity	13
3.17	Get Bottle Location	13
3.18	Register Bottle (Manual)	14
3.19	Register Bottle (Barcode Scanner)	14
3.20	Edit Bottle Brand	14
3.21	Edit Wine Color	15
3.22	Edit Wine Type	15
3.23	Edit Bottle Status	15
3.24	Edit Bottle Quantity	16
3.25	Edit Bottle Location	16
3.26	Delete Bottle	16
3.27	View Inventory	17
3.28	Search Bottle by Name	17
3.29	Filter by Brand	17
3.30	Filter by Color	18
3.31	Filter by Type	18
3.32	Sort Alphabetically (Brand)	18
3.33	Sort Alphabetically (Color)	19
3.34	Sort Alphabetically (Type)	19
3.35	Sort by Date	19

4	Packaging Requirements	21
4.1	Application Icon	21
4.2	Screenshots	21
4.3	Database Cleanup	21
4.4	User Manual	22
5	Performance Requirements	23
5.1	BAR CODE	23
5.2	DISTINGUISH BARCODES	23
5.3	Location Specific Entries	24
6	Safety Requirements	25
6.1	Consolidate a backup of all files	25
6.2	Database Security Practice	25
6.3	NEC Guidelines	25
7	Maintenance & Support Requirements	27
7.1	Maintenance and support	27
8	Other Requirements	28
8.1	Additional Features	28
9	Future Items	29
9.1	Additional Features	29

LIST OF FIGURES

1	The final product will be available on App Store and Google Play Store	6
2	Wine Inventory Sign in/Sign up/ Forgot Password Mock-up Designs	7

1 PRODUCT CONCEPT

Imagine that you are a wine and liquor lover, who is passionate about collecting different types of wine grapes from different regions. As the time goes by, the amount of wine and liquor bottles in your collection surprisingly increases and it is frustrating that you can't find the perfect bottle in your own cellar because you do not remember where the exact bottle is stored, in which bottle hole or which bin. In addition, we all know that the taste of the wine and liquor gets better as it ages. However, if a bottle has been opened, it gets put away in random places and can't be found when you want it.

1.1 PURPOSE AND USE

Expensive wine and liquor end up going past prime, gifted wine and liquor bottles get put in random bottle holes that can not be found easily. On the other side, if you own a small winery business, wine and liquor bottle inventory management could be time consuming and requires a decent amount of manpower by manually counting how many bottles are left in stock, which one is running out or you want to check which bottles have been opened and put back in the cellars. That brings us the opportunity to work on solving your problem with managing your cellar collections. It will bring you a better experience in finding your lovely bottle, check its condition.

Wine Inventory will be built to perform some tasks such as scan and add the wine and liquor bottles into the inventory with a specific cellar or bin location provided by the user along with its descriptions: color, origin, winery, vintage/year produced, styles: read/white/rose/sparking white/sparking rose. Its condition: opened or sealed, opened date, is it full, half or almost consumed. The app will help to mitigate the problems with locating the wine or liquor bottle, its condition, check to see if the bottle is getting expired(if the bottle is opened).

1.2 INTENDED AUDIENCE

Wine Inventory app will be made mostly for personal and general uses and small business size use only. The final product will be publicly available on App Store or Google Play Store with my purchase required. In addition, this app is particularly designed per the requirements from Christopher Conly PhD.



Figure 1: The final product will be available on App Store and Google Play Store

2 PRODUCT DESCRIPTION

This section provides the reader with an overview of Wine Inventory app. The primary operational aspects of the app is to help users maintain their wine selection base on their inputs. In addition, the app will coordinate user's inputs corresponding to the our database to extract more information about the bottle.

2.1 FEATURES & FUNCTIONS

The app will allow users to input their bottles information by scanning the bar code or if the bar code can't be found then users can manually input the information which includes winery or winemaker, wine name, primary grape varietal, year produced, style, a picture of it, especially it's storage location so user can easily find it, bottle's status (opened/unopened) as well as opened date. Afterwards, users will be able to have a list of bottles they have in storage, sort them by style, opened date, aging, etc...

2.2 EXTERNAL INPUTS & OUTPUTS

First of all, new users have to use their email and password to create an account with Wine Inventory. In addition, the main source of input comes from users by the bar code or the bottle's label. Then, the app will extract bottle's information such as bottle's name, data produced, winery type, wine style, origin. If the app cannot extract any data from the image then it will prompt users into input bottle's name, is it opened or unopened, is it consumed cold or regular temperature, the date bottle is opened, it's location(where user wants to put it). Secondly, in terms of output, the app will show users list of bottles they have inputted to the app database. Then, if user clicks a bottle, then the app will display the bottle information accordingly in columns and rows.

2.3 PRODUCT INTERFACES

Picture below are the app UI for Login, Sign up and Forgot Password mock up designs.



Figure 2: Wine Inventory Sign in/Sign up/ Forgot Password Mock-up Designs

3 CUSTOMER REQUIREMENTS

With our wine inventory app, we want to provide an easy-to-use, convenient, and efficient way for our customers to be able to keep track of their collection/stock. Our app allows the user to easily register any bottle of wine, set the amount of bottles, and provide a location for each individual wine. The convenience of having this app on their phone allows the user to always know what they have in stock and what they need to replenish.

3.1 GET USER FIRST NAME

3.1.1 DESCRIPTION

When first using the app, the user must create an account. The first step in creating an account is inputting their first name into the form.

3.1.2 SOURCE

Coding Avengers

3.1.3 CONSTRAINTS

N/A

3.1.4 STANDARDS

First name should be no more than 25 characters.

3.1.5 PRIORITY

Critical

3.2 GET USER NAME

3.2.1 DESCRIPTION

When first using the app, the user must create an account. The second step in creating an account is inputting their last name into the form.

3.2.2 SOURCE

Coding Avengers

3.2.3 CONSTRAINTS

N/A

3.2.4 STANDARDS

Last name should be no more than 25 characters.

3.2.5 PRIORITY

Critical

3.3 GET USER EMAIL

3.3.1 DESCRIPTION

When first using the app, the user must create an account. The third step in creating an account is inputting their email into the form.

3.3.2 SOURCE

Coding Avengers

3.3.3 CONSTRAINTS

N/A

3.3.4 STANDARDS

N/A

3.3.5 PRIORITY

Critical

3.4 GET USER PASSWORD

3.4.1 DESCRIPTION

When first using the app, the user must create an account. The fourth step in creating an account is inputting a password into the form.

3.4.2 SOURCE

Coding Avengers

3.4.3 CONSTRAINTS

N/A

3.4.4 STANDARDS

Password must be minimum 8 characters, containing at least:

- One uppercase letter
- One lowercase letter
- One number
- One special character (Ex: ! @ # \$ % & *)

3.4.5 PRIORITY

Critical

3.5 CREATE ACCOUNT

3.5.1 DESCRIPTION

When first using the app, the user must create an account. User is able to create an account once they have inputted all their information.

3.5.2 SOURCE

Coding Avengers

3.5.3 CONSTRAINTS

User may not create an account if there is an existing account for the inputted email.

3.5.4 STANDARDS

N/A

3.5.5 PRIORITY

Priority

3.6 VERIFY USER EMAIL

3.6.1 DESCRIPTION

After an account is created, the user's email needs to be verified for security purposes. Verification will be done through Firebase.

3.6.2 SOURCE

Coding Avengers

3.6.3 CONSTRAINTS

Since we are using Firebase for email verification, there is a possibility that there could be technical difficulties on that end when a user is verifying their email.

3.6.4 STANDARDS

User must verify email before having access to app.

3.6.5 PRIORITY

Critical

3.7 LOG IN USER

3.7.1 DESCRIPTION

After a user has created an account and verified their email, they can use their credentials to log in and have access to the app.

3.7.2 SOURCE

Coding Avengers

3.7.3 CONSTRAINTS

N/A

3.7.4 STANDARDS

N/A

3.7.5 PRIORITY

Critical

3.8 LOG OUT USER

3.8.1 DESCRIPTION

When a user is logged in the app, they are able to logout.

3.8.2 SOURCE

Coding Avengers

3.8.3 CONSTRAINTS

N/A

3.8.4 STANDARDS

N/A

3.8.5 PRIORITY

Critical

3.9 CHANGE USER PASSWORD

3.9.1 DESCRIPTION

After a user has created an account, they have the option to change their password. This will be updated in the database.

3.9.2 SOURCE

Coding Avengers

3.9.3 CONSTRAINTS

N/A

3.9.4 STANDARDS

Old password must be inputted and confirmed to create a new password. New password must meet password requirements.

3.9.5 PRIORITY

High

3.10 CHANGE USER NAME

3.10.1 DESCRIPTION

After a user has created an account, they have the option to change their first/last name. This will be updated in the database.

3.10.2 SOURCE

Coding Avengers

3.10.3 CONSTRAINTS

N/A

3.10.4 STANDARDS

New name must meet name requirements.

3.10.5 PRIORITY

High

3.11 DELETE USER

3.11.1 DESCRIPTION

After creating an account, the user has the option to delete their account. This would remove all of the user's information from the database.

3.11.2 SOURCE

Coding Avengers

3.11.3 CONSTRAINTS

N/A

3.11.4 STANDARDS

N/A

3.11.5 PRIORITY

Critical

3.12 GET BOTTLE BRAND

3.12.1 DESCRIPTION

When registering a bottle into the app, the first step is to input the brand of the bottle. (Ex: Stella Rosa, Domaine Leflaive, Castello Del Poggio)

3.12.2 SOURCE

Coding Avengers

3.12.3 CONSTRAINTS

N/A

3.12.4 STANDARDS

N/A

3.12.5 PRIORITY

Critical

3.13 GET WINE COLOR

3.13.1 DESCRIPTION

When registering a bottle into the app, the second step is to input the color of wine. (Ex: Red, White, Pink)

3.13.2 SOURCE

Coding Avengers

3.13.3 CONSTRAINTS

N/A

3.13.4 STANDARDS

N/A

3.13.5 PRIORITY

Critical

3.14 GET WINE TYPE

3.14.1 DESCRIPTION

When registering a bottle into the app, the third step is to input the type of wine. (Ex: Riesling, Zinfandel, Chardonnay)

3.14.2 SOURCE

Coding Avengers

3.14.3 CONSTRAINTS

N/A

3.14.4 STANDARDS

N/A

3.14.5 PRIORITY

Critical

3.15 GET BOTTLE STATUS

3.15.1 DESCRIPTION

When registering a bottle into the app, the fourth step is to set the bottle's status. (Ex: Open, Closed)

3.15.2 SOURCE

Source

3.15.3 CONSTRAINTS

Detailed description of applicable constraints...

3.15.4 STANDARDS

List of applicable standards

3.15.5 PRIORITY

Critical

3.16 GET BOTTLE QUANTITY

3.16.1 DESCRIPTION

When registering a bottle into the app, the fifth step is to input the quantity of bottles. (Ex: 1, 3, 12)

3.16.2 SOURCE

Coding Mavericks

3.16.3 CONSTRAINTS

N/A

3.16.4 STANDARDS

Quantity cannot be set to 0 during registration. May be 0 when editing quantity.

3.16.5 PRIORITY

Critical

3.17 GET BOTTLE LOCATION

3.17.1 DESCRIPTION

When registering a bottle into the app, the last step is to input the location of the bottle. (Ex: Slot 4, 1-3, B2)

3.17.2 SOURCE

Coding Avengers

3.17.3 CONSTRAINTS

N/A

3.17.4 STANDARDS

Since each user has a unique way of storing and labeling locations, the location input will be custom from the user.

3.17.5 PRIORITY

Critical

3.18 REGISTER BOTTLE (MANUAL)

3.18.1 DESCRIPTION

Once all the wine's information is inputted into the form, the user is able to register the wine into the app. This will be updated in the database.

3.18.2 SOURCE

Coding Avengers

3.18.3 CONSTRAINTS

Wine will not be registered if the same brand, color, and type is already registered (no duplicates).

3.18.4 STANDARDS

Each individual bottle will have a unique ID.

3.18.5 PRIORITY

Critical

3.19 REGISTER BOTTLE (BARCODE SCANNER)

3.19.1 DESCRIPTION

The user will be able to scan the barcode of the bottle with their phone's camera too automatically fill in bottle's information. The user will be able to edit any information scanned before registering the bottle.

3.19.2 SOURCE

Coding Avengers

3.19.3 CONSTRAINTS

We will be using an API to scan the wine barcodes and needs to be compatible with our project.

3.19.4 STANDARDS

N/A

3.19.5 PRIORITY

Moderate

3.20 EDIT BOTTLE BRAND

3.20.1 DESCRIPTION

After a bottle is done being registered, a user may edit the bottle's brand. This will be updated in the database.

3.20.2 SOURCE

Coding Avengers

3.20.3 CONSTRAINTS

N/A

3.20.4 STANDARDS

New bottle information will be cross-checked with the database to prevent any duplicates.

3.20.5 PRIORITY

High

3.21 EDIT WINE COLOR

3.21.1 DESCRIPTION

After a bottle is done being registered, a user may edit the bottle's color. This will be updated in the database.

3.21.2 SOURCE

Coding Avengers

3.21.3 CONSTRAINTS

N/A

3.21.4 STANDARDS

New bottle information will be cross-checked with the database to prevent any duplicates.

3.21.5 PRIORITY

High

3.22 EDIT WINE TYPE

3.22.1 DESCRIPTION

After a bottle is done being registered, a user may edit the bottle's type. This will be updated in the database.

3.22.2 SOURCE

Coding Avengers

3.22.3 CONSTRAINTS

N/A

3.22.4 STANDARDS

New bottle information will be cross-checked with the database to prevent any duplicates.

3.22.5 PRIORITY

High

3.23 EDIT BOTTLE STATUS

3.23.1 DESCRIPTION

After a bottle is done being registered, a user may edit the bottle's status. This will be updated in the database.

3.23.2 SOURCE

Coding Avengers

3.23.3 CONSTRAINTS

N/A

3.23.4 STANDARDS

N/A

3.23.5 PRIORITY

High

3.24 EDIT BOTTLE QUANTITY

3.24.1 DESCRIPTION

After a bottle is done being registered, a user may edit the bottle's quantity. This will be updated in the database.

3.24.2 SOURCE

Coding Avengers

3.24.3 CONSTRAINTS

N/A...

3.24.4 STANDARDS

N/A

3.24.5 PRIORITY

High

3.25 EDIT BOTTLE LOCATION

3.25.1 DESCRIPTION

After a bottle is done being registered, a user may edit the bottle's location. This will be updated in the database.

3.25.2 SOURCE

Coding Avengers

3.25.3 CONSTRAINTS

N/A

3.25.4 STANDARDS

New bottle location will be cross-checked with the database to prevent any duplicates.

3.25.5 PRIORITY

Priority

3.26 DELETE BOTTLE

3.26.1 DESCRIPTION

After a bottle is registered, a user may delete the bottle. This will remove the bottle from the database.

3.26.2 SOURCE

Coding Mavericks

3.26.3 CONSTRAINTS

N/A

3.26.4 STANDARDS

N/A

3.26.5 PRIORITY

Critical

3.27 VIEW INVENTORY

3.27.1 DESCRIPTION

When the user wants to view their inventory, the app will display all bottles in the database.

3.27.2 SOURCE

Coding Avengers

3.27.3 CONSTRAINTS

N/A

3.27.4 STANDARDS

N/A

3.27.5 PRIORITY

Critical

3.28 SEARCH BOTTLE BY NAME

3.28.1 DESCRIPTION

When viewing inventory, user can search by name using the search bar.

3.28.2 SOURCE

Coding Avengers

3.28.3 CONSTRAINTS

N/A

3.28.4 STANDARDS

N/A

3.28.5 PRIORITY

High

3.29 FILTER BY BRAND

3.29.1 DESCRIPTION

When viewing inventory, user can filter results by brand name.

3.29.2 SOURCE

Coding Avengers

3.29.3 CONSTRAINTS

N/A

3.29.4 STANDARDS

N/A

3.29.5 PRIORITY

High

3.30 FILTER BY COLOR

3.30.1 DESCRIPTION

When viewing inventory, user can filter results by wine color.

3.30.2 SOURCE

Coding Avengers

3.30.3 CONSTRAINTS

N/A

3.30.4 STANDARDS

N/A

3.30.5 PRIORITY

High

3.31 FILTER BY TYPE

3.31.1 DESCRIPTION

When viewing inventory, user can filter results by wine type.

3.31.2 SOURCE

Coding Avengers

3.31.3 CONSTRAINTS

N/A

3.31.4 STANDARDS

N/A

3.31.5 PRIORITY

High

3.32 SORT ALPHABETICALLY (BRAND)

3.32.1 DESCRIPTION

When viewing inventory, user can sort results alphabetically by the brand.

3.32.2 SOURCE

Coding Avengers

3.32.3 CONSTRAINTS

N/A

3.32.4 STANDARDS

N/A

3.32.5 PRIORITY

High

3.33 SORT ALPHABETICALLY (COLOR)

3.33.1 DESCRIPTION

When viewing inventory, user can sort results alphabetically by the color.

3.33.2 SOURCE

Coding Avengers

3.33.3 CONSTRAINTS

N/A

3.33.4 STANDARDS

N/A

3.33.5 PRIORITY

High

3.34 SORT ALPHABETICALLY (TYPE)

3.34.1 DESCRIPTION

When viewing inventory, user can sort results alphabetically by the brand.

3.34.2 SOURCE

Coding Avengers

3.34.3 CONSTRAINTS

N/A

3.34.4 STANDARDS

N/A

3.34.5 PRIORITY

High

3.35 SORT BY DATE

3.35.1 DESCRIPTION

When viewing inventory, user can sort results by the date the bottle was registered.

3.35.2 SOURCE

Coding Avengers

3.35.3 CONSTRAINTS

N/A

3.35.4 STANDARDS

N/A

3.35.5 PRIORITY

High

4 PACKAGING REQUIREMENTS

As an app, there is no physical packaging for the product, but there are still some requirements for presenting the app in app stores.

4.1 APPLICATION ICON

4.1.1 DESCRIPTION

The app will have a logo on the device's home screen. The icon image files can be made manually with or with automated tools like [AppIcon.co](#) or Android Studio's [Image Asset Studio](#).

4.1.2 SOURCE

Team member (Greg)

4.1.3 CONSTRAINTS

Another team member will be designing the logo and the app will have a placeholder icon during development until they're done.

4.1.4 STANDARDS

The icon should adhere to Apple's (iOS and iPadOS) and Google's (Android) human interface guidelines including, but not limited to, resolution and size, colors, and shape. For documentation, see Apple's Human Interface Guidelines: [App Icon & Image Size and Resolution](#) and Android Developers: [Google Play icon design specifications](#).

4.1.5 PRIORITY

High

4.2 SCREENSHOTS

4.2.1 DESCRIPTION

Screenshots of the finished app with example data should be captured for demonstration and presentation.

4.2.2 SOURCE

Team member (Greg)

4.2.3 CONSTRAINTS

N/A

4.2.4 STANDARDS

We should adhere to typical dimensions. See guidelines for [iOS screenshots](#) and [Android screenshots](#).

4.2.5 PRIORITY

Moderate

4.3 DATABASE CLEANUP

4.3.1 DESCRIPTION

Any development or testing data in Firebase should be cleared or at least made inaccessible prior to launch.

4.3.2 SOURCE

Team member (Greg)

4.3.3 CONSTRAINTS

N/A

4.3.4 STANDARDS

N/A

4.3.5 PRIORITY

High

4.4 USER MANUAL

4.4.1 DESCRIPTION

The app will have a user manual/help section embedded.

4.4.2 SOURCE

Project Charter

4.4.3 CONSTRAINTS

N/A

4.4.4 STANDARDS

N/A

4.4.5 PRIORITY

High

5 PERFORMANCE REQUIREMENTS

The application shall be able to scan user intended barcode within a 2-4 seconds. The process of the inventory management can be done manually and the connection can be established with the stock in the database. The additional details can be entered manually along with a picture of the wine bottle. The setup process for storing the information should not take more than 1-3 minutes.

5.1 BAR CODE

5.1.1 DESCRIPTION

The inbuilt barcode scanner shall be able to scan the barcode accurately with minimum number of errors.

5.1.2 5.1.2 CONNECT TO THE DATABASE

After the product is scanned or manually entered, the connection with the beverage inventory is established and the data is saved in the database.

5.1.3 SOURCE

Coding Avengers Team

5.1.4 CONSTRAINTS

5.1.4.1 Internet connection is required to connect to the database 5.1.4.2 Lighting that may modify readability of the barcode.

5.1.5 STANDARDS

Database shall store all the information.

5.1.6 PRIORITY

High

5.2 DISTINGUISH BARCODES

5.2.1 DESCRIPTION

The inbuilt scanner shall distinguish the barcode which the user scans and lets the user decide on the editing part.

5.2.2 SOURCE

Coding Avengers Team

5.2.3 CONSTRAINTS

The lighting and the camera condition can effect the the use of the scanner.

5.2.4 STANDARDS

N/A

5.2.5 PRIORITY

Moderate

5.3 LOCATION SPECIFIC ENTRIES

5.3.1 DESCRIPTION

The app shall allow the end user to name the space along with the exact location the beverage is stocked. The other information along with the location is added like if the bottle is opened and needs to be finished earlier.

5.3.2 SOURCE

Coding Avengers Team

5.3.3 CONSTRAINTS

The end user input about the location might not be precise or correct for the reference.

5.3.4 STANDARDS

N/A

5.3.5 PRIORITY

High

6 SAFETY REQUIREMENTS

The Wine Inventory project is a software application development project and does not have any safety concerns like exposure to toxic chemicals, sharp edges that could harm a user, no breakable glass in the enclosure, no direct eye exposure to infrared/laser beams, packaging/grounding of electrical connections to avoid shock, etc. However, the safety measurements like files and database backup, wiring compliance, etc. is taken into consideration.

6.1 CONSOLIDATE A BACKUP OF ALL FILES

6.1.1 REQUIREMENT

To ensure the complete safety in case of data loss, backup will be present to keep the work going on smoothly. Moreover, an archive of data is considered for prevention of any kind of data loss.

6.1.2 SOURCE

Coding Avengers Team

6.1.3 CONSTRAINTS

Prone to hash collisions

6.1.4 STANDARDS

Merkle tree data structure

6.1.5 PRIORITY

Critical

6.2 DATABASE SECURITY PRACTICE

6.2.1 REQUIREMENT

A web server might be likely to be attacked as it is publicly accessible. If the web server is compromised, the database server might be compromised if it runs on the same machine. Hence, to ensure the safety web applications and database firewalls shall be used to protect from database security threats by a firewall.

6.2.2 SOURCE

CSE Senior Design Laboratory Policy

6.2.3 CONSTRAINTS

User Data Constraint Specifying an authorization constraint Specifying a Web Resource Collection

6.2.4 STANDARDS

Database Server Security Standard (DSSS)

6.2.5 PRIORITY

Critical

6.3 NEC GUIDELINES

6.3.1 REQUIREMENT

The equipment listed or labelled in the NEC guidelines code must be installed and used per instructions in the listing or labelling requirements. The installation of the electric devices will be considered to be safe considering the circuit nominal system voltage making sure not to exceed the rating of the equipment.

6.3.2 SOURCE

CSE Senior Design Laboratory Policy

6.3.3 CONSTRAINTS

Certain systems cannot be grounded

6.3.4 STANDARDS

The National Electric Code(NEC)

6.3.5 PRIORITY

Critical

7 MAINTENANCE & SUPPORT REQUIREMENTS

7.1 MAINTENANCE AND SUPPORT

7.1.1 DESCRIPTION

Developers shall use online free software maintenance tools during entire phase of design and development to minimize error. A QA team shall test the product for any errors before send to the end user. Bugs shall fix by the developers after screening the errors in the line of code. There will be telephone and online technical trouble shooting team. User manual will be provided for further assistance. Software shall update frequently as required. Additional features shall add with additional cost and API documentation shall provide online. Feedback and help desk shall available.

7.1.2 SOURCE

Coding Avengers Team,Maintenance Group

7.1.3 CONSTRAINTS

Maintenance teams need to set up for the evolution of app.

7.1.4 STANDARDS

App will be compatible for both android and iPhone platform

7.1.5 PRIORITY

High

8 OTHER REQUIREMENTS

8.1 ADDITIONAL FEATURES

There are not all features are added in app. It is only for personal and small business use. App can add features to be usable for big business.

8.1.1 DESCRIPTION

There will be software maintenance and support services include long term and pay-as you-go. App shall add additional features in window, Linux, Unix, Mac OS in the future.

8.1.2 SOURCE

Coding Avengers team

8.1.3 CONSTRAINTS

Maintenance will be done free cost for certain time period and additional cost will apply.

8.1.4 STANDARDS

App will be compatible with other platform like window, Linux, Unix, Mac OS.

8.1.5 PRIORITY

Future

9 FUTURE ITEMS

9.1 ADDITIONAL FEATURES

There are not all features are added in app. It is only for personal and small business use. App can add features to be usable for big business.

9.1.1 DESCRIPTION

There will be software maintenance and support services include long term and pay-as you-go. App shall add additional features in window, Linux, Unix, Mac OS in the future.

9.1.2 SOURCE

Coding Avengers team

9.1.3 CONSTRAINTS

Maintenance will be done free cost for certain time period and additional cost will apply.

9.1.4 STANDARDS

App will be compatible with other platform like window, Linux, Unix, Mac OS.

9.1.5 PRIORITY

Future

REFERENCES