Requirements Document

CSCI-310 Software Engineering

Professor: **Nenad Medvidović**

TA: **Sarah Cooney**

Fall 2017

$a₦i₮¥

Ba$iL (TEAM 15)

Tri Nguyen 1410884747

Ang Li 3340862395

Utsav Thapa 9717428024

Andre Takhmazyan 8764629970

Kevin Nguyen 215820474

2. Preface

This specification document is intended for both technical and non-technical readers. Sections 2-5 are non-technical while sections 6 and 7 require some general knowledge of software programming. This document outlines the functional and nonfunctional requirements of the android application “Sanity.” This document is intended for stakeholders, software engineers, and end users of “Sanity” and serves as a guideline of what the application should and should not do.

Version History:

9/15/17 v0.9
Requirements Document Started

10/16/17 v36.69
Requirements Document Revision #1 - Post Implementation

3. Introduction

Budgeting can be difficult, but having an android app to keep track of spending can make it easier. $anity is an android application that helps users budget their spending across a variety of categories. Users can create a budget for a certain spending category over a period of time (e.g. groceries, gas, dining dollars) and set a limit of maximum spending in that category. When the user makes a purchase, he/she will be able to enter the amount and category of the spending into $anity, and $anity will deduct the amount from the current budget. When the user approaches the limit for any of their budget (e.g. 75% of the budget) they will receive notifications that they are approaching their budget limit. This application is similar to the android program “Mint” in which both show multiple budgets and and you can add and remove budgets. Our program also shows analytics which Mint does not.

4. Glossary

User : Anyone who purchases and uses $anity.

Budget : Maximum spending amount for each category set by the user that the app keeps track of.

Category : Classifications created by the user or the app to divide the budget in parts.

Transaction : when the user makes a purchase and records the amount in $anity.

Analytics : A visual representation of current and past budget data (usually as a line graph)

5. User Requirements Definition

User Requirements

1. $anity will track user purchases and returns from multiple categories.
2. $anity will allow users to see graphs and charts of their purchases for selected time period.
3. $anity will allow users to set a maximum amount for specified budget.
4. $anity will allow users to create budget categories but also provide common presets.

a. Scenario/user story #1

A user buys a coffee for $3 from Starbucks using a credit card. He uses the app to enter in $3 in the food category. The app deducts $3 from his monthly food budget. Since this does not put the user over budget or exceed 75% of the food budget, the app sends no notifications.

b. Scenario/user story #2

The user buys camping supplies for $300 but he does not have a category where he can add the purchase to. He adds a new budget category for camping and

sets it to $1000 a year, and adds the transaction for the camping supplies to it. His remaining camping budget for the year is now $700.

c. Scenario/user story #3

The user buys a t-shirt from the mall for $20 but when he gets home he finds out it doesn't fit. He has already added the amount of the purchase to his clothes category but wants to return the t-shirt. After the return, he adds a new purchase amount with a negative value that lowers his total spending in the clothes category.

d. Scenario/user story #4

The user purchases a DSLR camera priced at $4000, however his/her budget for technology purchases is limited to $300. He/she should receive a one time notification noting his/her instance of **high spending** and if there are any subsequent purchases that fall in the technology category, he/she should receive a one time notification noting his/her instance of **over budget spending**.

e. Scenario/user story #5

The user is approaching his total spending for the month, reaching $430 out of $500 of his overall budget. The user wants to see where his/her spending has gone toward the most, and looks at the home screen of $anity to see the pie chart of their total spending for the month. It turns out that 50% of their spending has gone towards gas, and the user can now decide to cut down on gas spending for the rest of the month.

6. System Requirements Specification

a. Functional Requirements

1.1 $anity will track user purchases and returns from multiple categories.

1.11 The user data will be stored locally on the physical device via *SQLite*.

~~1.12 To process a return, the user will enter a new purchase with negative value.~~

1.12 To process a return, the user will click on an add refund button to enter the refund amount.

1.13 When a transaction is added, users can fill in the amount manually.

1.14 When a transaction is created, users can add an optional memo.

1.15 Transactions are added to the appropriate category by clicking the Add Transaction button within the appropriate category.

1.2 $anity will allow users to see graphs and charts of their purchases for selected time period.

1.21 The user will have a home screen in which they can view all of their budget categories and the amount, percentage, and period of time remaining until the next budget cycle for each category.

1.22 The user will be able to view total budget as a pie chart. Other analytics can be seen within the budget categories such as ~~total spent~~ transactions over time.

1.23 The user will be able to see graphs of their overall transactions for the last month and their transactions per budget for at least the last six budget periods.

1.3 $anity will allow users to set a maximum amount and budget period for specified budget.

1.31 Every time a new purchase is added, the app checks the budget against the user set limit.

1.32 $anity will notify users if they are above 75% of their budgeted category. The user may also change the threshold to a different percent.

1.33 $anity will notify users if they are above their budget limit.

~~1.34 The user will have an option to roll over budget surpluses to the next month. By default, surpluses will not roll over.~~

1.34 The user will be able to set a budget period for each budget, the budget period changes will take effect immediately and budgets will reset at the end of the period.

1.35 The user will also be able to delete budgets. When a budget is removed, the budget limit is deducted from the overall budget.

1.36 The threshold for the budget can be changed by the user, these changes will be reflected immediately in the main menu.

1.4 $anity will allow users to create budget categories but also provide common presets.

When the user sets up his/her budget for the first time, common budget categories such as Housing, Food, and Transportation will be provided. There will also be an option to create custom category.

b. Non-functional Requirements

2.1 The app will not require excessive internet data to function after being downloaded**.**

2.2 The app download will be less than 200 MB.

2.3 The app will be created using Android Studio.

2.3 Internet connection to download the app, minimum dial-up

2.4 Any number of transactions can be made on different phones without crashing.

2.5 Any number of users can remain online at the same time without crashing.

2.6 The app will keep user passwords and security answers secure by storing hashes rather than user-entered values.

2.7 The app will be user friendly and intuitive to operate without the need of a user manual.

2.8 The app will not cause any excessive battery drain.

2.9 The app will be fault tolerant and will crash less than once per 20 transactions.

c. UI Appearance Requirements

~~3.1 When the user presses the button to open the app, they are greeted with a intro screen with the app name.~~

3.1 The next screen will show all budget categories and and how far each one is to being over budget.

3.2 The user can press a button to add a category, a button to edit each category, and a button to add a purchase to each category.

3.3 The user can also press a button to view analytics of past budget months.

3.4 The phone will also display notifications when a budget category reaches certain thresholds.

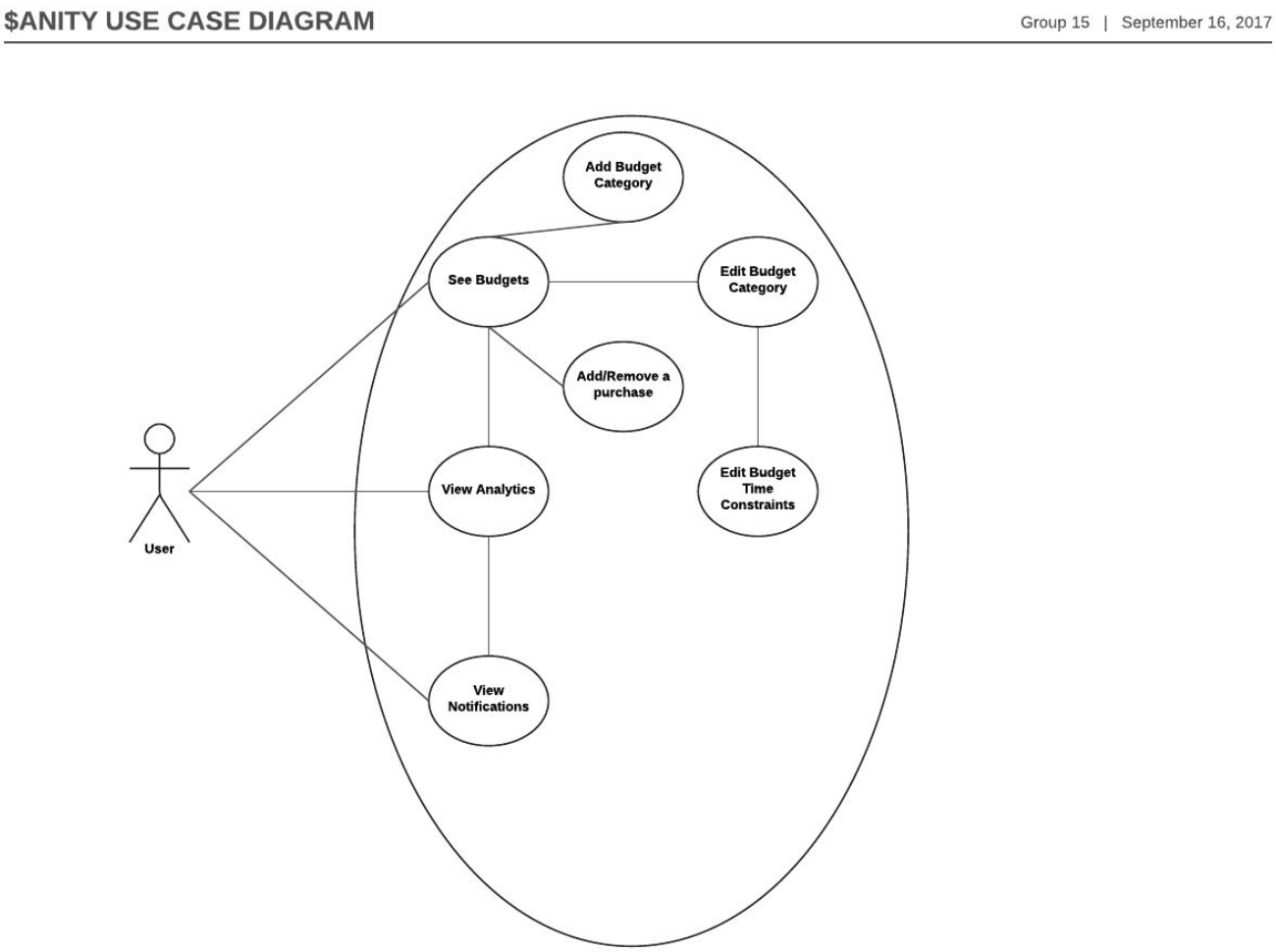
3.5 The app will be color themed in Android Material Design

3.6 The app will be pleasing to the eye and will not cause eye strain

3.7 The app will not feature any flashing lights

7. System Models

Use Case Diagram



The Total Budget Equation

$$TotalBudget = (\sum_{0}^{n} budget_i) + MiscellaneousExpenses$$

The total budget can be accurately modeled as the sum of all budget categories along with any miscellaneous Expenses that can't be assigned to a pre-existing

OR custom category.

8. Index