Software Requirement Specification version 1.0

Table of Contents

[1. Introduction 1](#_Toc302044349)

[1.1 Project Overview:- 1](#_Toc302044350)

[1.2. Scope and Limitations:-](#_Toc302044351) 2

[2. Use Cases 3](#_Toc302044352)

[Use case for System Setting Password : 4](#_Toc302044354)

[Use case to Maintaining Expense Type: 4](#_Toc302044355)

[Use case to Report Generation: 7](#_Toc302044356)

[Use case to Maintaining Account: 8](#_Toc302044357)

[Use case to Change Password: 10](#_Toc302044357)

[3. Dependencies and Risks:- 10](#_Toc302044359)

[3.1 Dependency:- 10](#_Toc302044360)

[4. Software and Hardware Requirements:- 11](#_Toc302044362)

[4.1 System Requirements:- 11](#_Toc302044363)

5. Supported Platforms:-……………………………………………………………………..12

[6. Installation :- 12](#_Toc302044370)

#### 

#### 1. Introduction

Mobile phones are tremendously changing the trend of markets. Mobiles are not only used for communication but also used to make things easier to save the time etc. Many applications are being developed in smart phones to complete many complex tasks in much convenient way. One such application is “Mobile Account Management” System.

Mobile apps are convenient. They're quick. And they let you access the information you need to know, whenever you want to know it. Mobile apps are a great way to stay in touch with your firm no matter where you happen to be. The application Mobile Account Management is an application developed for android based Operating System employed in mobile phones. This application makes easy to maintain the daily expenses for the people like housewife and salesman.

#### Project Overview

Mobile account gains more popularity as a number of mobile device users are on the increase the ways of doing businesses and marketing have tremendously changed from the past. Mobile Account Manager makes daily expense account maintenance easy. It is used to store the expense details of a user. Using this app the user can enter his daily expense information like a shopping, purchasing goods expense details etc. This project facilitates the user to enter a particular kind of expense information; later the user can view and generate a report on expense details.

Using Mobile Account Manager, a user can maintain expense account information i.e. this facilitates for the user to store daily expenses in a category manner and also the user can generate reports. Users of this application are from housewife to a sales man. This application is not only facilitate the user to generate reports but also the user can mail that report to others (in the case of sales man can mail the report to the manager).

#### 1.2 . Scope and Limitations

## SCOPE:

The following are the features of this application:

* This application provides more security for the information of users by login mechanism through a valid password.
* This application allows the user to enter his/her daily category wise expense details.
* By this application the user can add new category of expense to this application.
* The user can able see the total expense of a particular category daily wise, weekly wise and monthly wise also
* This application allows for changing the password for accessing this application, if he remembers his previous password.

## LIMITATIONS:

Some of the pitfalls in the application are as follows:

* Inorder to be more interactive ,it needs to be developed in using both web and android applications.
* This application does not provide the balance sheet for a user account.
* Not highly scalabe , it needs integration .
* Mailing report to the another person is not implemented.

#### Use Cases

Usecase diagram is one ampong the 9 UML diagrams. The usecase diagrams provides us the pictorial representation of each and every individual involved in the application.It involves actors and their actions .

The basic UseCase diagram of “MOBILE ACCOUNT MANAGEMENT” application is shown in figure1.



Figure1

## Use Case Descriptions:

The brief descriptions of the actions performed by an actor are as follows:

### Use case for Setting System Password:

**Description**  : This used for setting the system password or the

application password for accessing this. User need to

set the password for one time, and he should give the

password for next time access.

**Actor** : Operator.

**Expected Input** : User needs to fill the Password field and Confirm

Password field , then press ok Button.

**Expected Output** : User can able to see A Toast message as “Password

and Confirm Password did not matched Please Enter

Again” if, he enter wrong password otherwise, he can

able to see the Home screen.

### Use case for Maintaining Expense Type:

**Description**  : This used for maintaining Expense Type, User can able

to add a new expense type, view a existing expense ,

update and delete also.

**Actor** : Operator.

**Expected Input** : User needs to open the application by entering the

password to the system..

**Expected Output** : A new screen comes out, containing three button such

as Report, AddnewType and ChangePassword .

### Use case for Add a New Expense Type

**Description**  : This used for adding a new expense type for putting the

expenses in that category.

**Actor** : Operator.

**Expected Input** : User needs to click on AddNew Type Button, then fill

the expense type and description and finally click on

“Store Button”.

**Expected Output** : If he click on Store without entering any value, the

one Toast message will be shown as “Type Must Not

Be Empty” , with entering some data it a Toast

message will be shown as “ Store successful” in

the same screen.

### Use case for Delete Expense Type:

**Description**  : This used for delete a Expense Type.

**Actor** : Operator.

**Expected Input** : User needs to check one or multiple Expense type to

delete, click Menu Button in the device and click on

delete, then click on Yes if he want to delete otherwise

click on NO.

**Expected Output** : If he click on No a Toast message will be shown as

“no record is deleted”, If he click on “Yes” it deleted

the selected and back to the previous screen , where a

list of Expense type will be displayed without the

deleting one or multiple.

### Use case for Update a Expense Type:

**Description**  : This used for Update a Expense type that includes

Type of the expense and description.

**Actor** : Operator.

**Expected Input** : User needs to check one Expense Type to update, click

Menu Button in the device and click on update

fill the required field to which you want to

update then click on Update , if you he want to delete

otherwise, click on Back in the device.

**Expected Output** : If he click on Update a Toast message will be shown

as “updating successful ”, If he click on Back in the

device it returns back to the previous screen .

### Use case for View a Expense Type:

**Description**  : This used for View a Expense type that includes

Type of the expense and description.

**Actor** : Operator.

**Expected Input** : User needs to check one Expense Type to view, click

Menu Button in the device and click on view.

**Expected Output** : If will display the name of the Expense Type and

how many expenses added to that type, with date and

amount.

### Use case for Report Generation:

**Description**  : This used for Report generation for the entire

application as well as for individual Expense type.

**Actor** : Operator.

**Expected Input** : User needs to “Report” button in the home screen.

**Expected Output** : A will display a new screen for displaying the report

and that contains two button for selecting two date

### Use case for Report Generation for Application:

**Description**  : This used for report generation for the entire

application.

**Actor** : Operator.

**Expected Input** : User needs to “Report” button in the home screen,

then choose two date one is from date and another is

two date, then click on “Generate Report”.

**Expected Output** : A will display the total amount expenses and also

Categorize total expenses in a list manner.

### Use case for Maintaining Account:

**Description**  : This used for maintaining Account for individual

Expense Type. User can able to add a new expenses

with date .

**Actor** : Operator.

**Expected Input** : User needs to check one Expense Category, then Click

on Menu Button in the device and Click on view.

**Expected Output** : If will display the name of the Expense Type and

how many expenses added to that type, with date and

amount.

### Use case for Add Amount:

**Description**  : This used for adding the amount of expense for that

Expense type.

**Actor** : Operator.

**Expected Input** : User needs to click “AddNew” Button, then enter the

amount then Click on save.

**Expected Output** : If he click on Save without entering amount, then

one Toast message will be shown as “Must Enter

Amount” , and returns back to the previous screen.

### Use case for Delete Amount:

**Description**  : This used for deleting the amount of expense for that

Expense type.

**Actor** : Operator.

**Expected Input** : User needs to check that amount, Click on menu

Button, Click on Delete then click on “OK”.

**Expected Output** : If he click on “Ok” that amount will be deleted from

that Expense type, If he click on “No” that amount

will not be deleted from that Expense type .

### Use case for Update Amount:

**Description**  : This used for updating the amount of expense for that

Expense type.

**Actor** : Operator.

**Expected Input** : User needs to check that amount, Click on menu

Button, Click on Update then update the amount how

much you want, then click on “Update” button.

**Expected Output** : If he click on “Update” that amount will be update

and it will be shown in the previous screen.

### Use case for Change Password:

**Description**  : This used for changed password, that is used for

accessing this application.

**Actor** : Operator.

**Expected Input** : User needs to click on Change Password and Enter the

old password, new password and confirm password

for new one then click on submit.

**Expected Output** : If user enters correct old password new password and

confirm password , then it will show a Toast message

as “password changed successfully” , otherwise entered

# 3. Dependencies and Risks:-

## 

## 3.1 Dependency:-

This application depends on the memory of the user, if he remember the application password, then only he can use that application, otherwise this application does not open. If the application implement the features like sending mail to other, then Internet connection is necessary.

# 

# 4. Software and Hardware Requirements:-

|  |  |  |
| --- | --- | --- |
| **S/W Requirements** | **:** | ***Supported Operating Systems***   * Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit) * Mac OS X 10.5.8 or later (x86 only) * Linux (tested on Ubuntu Linux, Lucid Lynx) * GNU C Library (glibc) 2.7 or later is required. * On Ubuntu Linux, version 8.04 or later is required.   **Supported Development Environments**   * Eclipse IDE * Eclipse 3.5 (Galileo) or greater * Note: Eclipse 3.4 (Ganymede) is no longer supported with the latest version of ADT. * Eclipse JDT plugin (included in most Eclipse IDE packages) * Eclipse IDE for Java Developers * Eclipse Classic (versions 3.5.1 and higher) * Eclipse IDE for Java EE Developers * JDK 6 or JDK 7(JRE alone is not sufficient) * Android Development Tools plugin (recommended)s |
| **H/W Requirements** | **:** | |  |  |  | | --- | --- | --- | | ***Component type*** | ***Approximate size*** | ***Comments*** | | SDK Tools | 35 MB | Required. | | SDK Platform-tools | 6 MB | Required. | | Android platform (each) | 150 MB | At least one platform is required. | | SDK Add-on (each) | 100 MB | Optional. | | USB Driver for Windows | 10 MB | Optional. For Windows only. | | Samples (per platform) | 10M | Optional. | | Offline documentation | 250 MB | Optional. | |

**5. Supported Platforms:-**

This application is supported for the following platforms:-

* Linux
* Windows XP
* Vista

This application is developed to support all platforms

This system will work in Linux which is a best operating system as per the security concern.

# 6. Installation and Deployment:-

## 

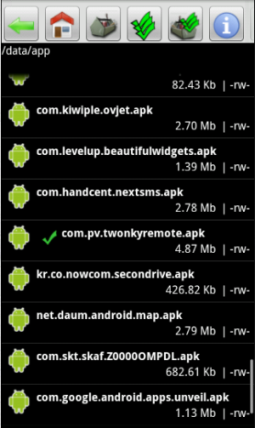
## Installation:-

You need a **file manager in android mobile**. The best file manager I found in android market is [Astro](https://market.android.com/details?id=com.metago.astro&feature=search_result). This can be used in both rooted and **non-rooted android mobiles**. For rooted android users you can go to root explorer. It has lot of options in exploring your mobile phone. Astro is a free version but root explorer is a paid version. Let me explain you the steps in **installing .apk files** in your android phone.

**1.** Copy the **apk file to your android phone** or download the file using your mobile phone browser.

**2.** Go to your file manager and start searching the name of your **.apk file**.

**3.** Locate your file and tap on the file to **install in your android phone**.



**4.** When you tap the file you have two options before you. One is open app manager and other one is app browser.

**5.** Use app manager to install the file in your android phone.

**6.** That’s all your .apk file is installed in your android phone.