

­­***Capstone Project Document***

*Personal Finance Management*

|  |  |
| --- | --- |
| **PFM Team** | |
| **Group Members** | Lại Lê Lệ Linh – LinhLLL01355  Nguyễn Đức Lộc – LocND01006  Nguyễn Anh Tuấn – TuanNA01030  Lê Thị Minh Khánh – KhanhLTM01262  Nguyễn Thị Gấm – GamNT01326 |
| **Supervisor** | Teacher Nguyễn Văn Hiển |
| **Ext Supervisor** |  |
| **Capstone Project code** | PFM |

HaNoi, 25thNov, 2012

**Record of changes:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Change Item** | **Description** | **By** | **Version** |
| 25/11/2012 | All | Create the document | GamNT01326 | 0.1 |
|  |  | Edit some content |  | 0.2 |
|  |  | Edit some content |  | 0.3 |
|  |  | Edit some content |  | 1.0 |

# Introduction

## Project overview

### Project information

* Project name: **Personal Finance Management**
* Project code: **PFM**
* Project type**: Mobile application**
* Duration: **Sep 2012 to Dec 2012**

### Group information

Supervisor:

Name: **Mr. Nguyen Van Hien**

E-mail: [**hiennv@fpt.edu.vn**](mailto:hiennv@fpt.edu.vn)

Phone: **0977.999.707**

Role: **Lecture**

Team members:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | **Full name** | **Student code** | **Phone** | **E-mail** | **Role** |
| 01 | Nguyen Anh Tuan | 01030 | 01229249209 | TuanNA01030@fpt.edu.vn | Leader |
| 02 | Lai Le Le Linh | 01355 | 01689908772 | LinhLLL01355@fpt.edu.vn | Member |
| 03 | Le Thi Minh Khanh | 01262 | 01665291990 | KhanhLTM01262@fpt.edu.vn | Member |
| 04 | Nguyen Thi Gam | 01326 | 01685201603 | GamNT01326@fpt.edu.vn | Member |
| 05 | Nguyen Duc Loc | 01006 | 01234351990 | LocND01006@fpt.edu.vn | Member |

## Initial idea

The idea of our product comes from actual problems, that how to help people manage their personal finance in the way of the most quickly and usefully. Normally, you have to save your receipts, record all your purchases, add up your expenses, and calculate how much you have left to spend or save. Maybe that final result sends you into unimaginable even complete shock, and you wonder why your money goes so quickly. You must spend lots of time for recalling accounts payable, expense and income. Sometime, you forgot and you cannot remember all of them.

And now we’d like to introduce our product which will bring to you a place to easily manage your finance, without waste of time to remember and don't afraid forget any account. It will do all of work for you: automatically calculating your expenses and purchases; analyzing your spending with reports and pie charts; alerting you of upcoming bills; planning your future spending; and more.

Not only manages finance, our system but also helps storage data by synchronization data. With this when you change or lost mobile, all your data will not be lost and you can use this function to get back data and continue use application but don’t need input again.

With the rapidly expansion of mobile with android OS, we hope that our product will become most popular application, to help more people managing personal finance.

## Overview of existing systems

There are some existing system and our analysis about them.

### Application in Vietnam

#### Money Lover

* **Features:**

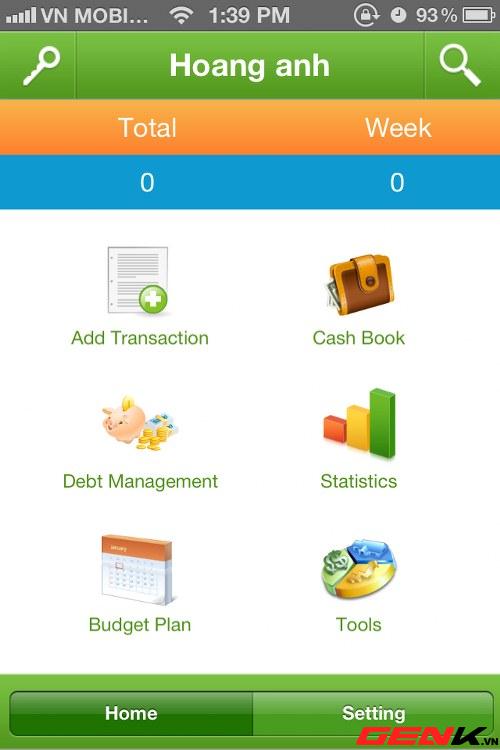
This application was developed by a Vietnamese. Money Lover is a financial budget management app that can be used for simple tasks such as tracking income and expenses on up to smarter advanced tools such as debt management and budget reports. The application supports more than 30 languages.  
After successful installation, the main interface of the program, including six function buttons will appear.

* Add Transaction: button will help you record the transaction revenues, expenditures, loans, investment...
* Cash Book: is a list of income and expenditures
* Debt Management: helps you manage the lending and borrowing easier.
* Statistics: give you charts and compare the income and expenditures.
* Budget plan: where to plan, estimate your expenses.
* Tools, gives you four basic tools: Currency Exchange, Bank Finder, ATM Finder and Interest Rate.

The app supports multiple profiles (in Premium), for example, you can set up one profile for personal and one for your office. It also can protect your data when you go to the settings to enable “Pattern Lock”. More importantly, have a look at the Auto Backup feature. From there you can set up scheduled backups of your financial data.

When you have an income or expenditure, you add them to "Cash Book" by selecting "Add transaction" and choose types (income, expense, Receivable or Payable). Then input name, select a category ... Finally, choose "Save" button, your income and expenditures are recorded and managed very easily. To view, you touch "Cash Book".

* **Advantages**
* Loaded with features both simple and comprehensive for personal financial management!
* Support for pattern lock to open app and automatic schedule of data backup!
* Produce charts and stats from data.
* Supports multiple profiles (in Premium version).
* Support many functions and languages
* **Disadvantages**
* Does not have a synchronize solution.
* Have too many unnecessary functions such as “Currency exchange”, “Tip calculator”, “ATM finder”, “Bank finder”.
* Have to buy the full version of the application to use some functions like “Account manager”…
* Some functions have not completed yet and need to be updated in the next version. It is also uncomfortable to bring user to home screen, because user must go back to many screens.



*Figure 1: Money Lover Interface*

### Foreign applications

#### CWMoney

* **Features**

This application helps user to keep track of all their expenses and incomes in an easy way.

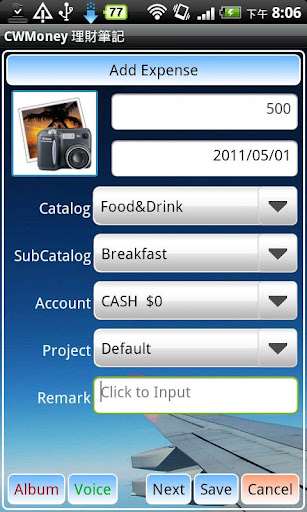
First of all, you need to select your language and then restart the application in order to make it work properly. Then, you can start use apps by adding some expenses or incomes.

The man interface of program including four functions:

* Expense: you can make a recording, set a default picture, and include a category and subcategory (Food and Breakfast for instance), the type of the account used (Cash, Credit Card…) and even the project (Holiday, Wedding, Travel…). You can also take notes about any particular thing you want to add.
* Income: same Expense function
* Data Manager: keep track of the money spent in a great variety of ways. A report is also provided (in a Pie or a Bar Chart) to help you visualize everything better.

Bottom of the screen shows total expense, income follow daily, weekly and monthly.

* **Advantages**
* The system is simple and easy to understand for users.
* Support UI Theme.
* Support multiple accounts and currency.
* Support GPS, Camera, Album, Voice.
* Data Export (Excel XML, CSV).
* Backup Send to Dropbox, Gmail, FTP...
* **Disadvantages**
* Have less function, lack schedule function, borrowing and lending function.
* Must purchase to synchronize data.
* Have some items are unnecessary for users.



*Figure 2: CWMoney Interface*

#### Easy Money

* **Features**

This application helps you track your expenses and stick to your budget quickly and easily using a simple yet powerful user interface.

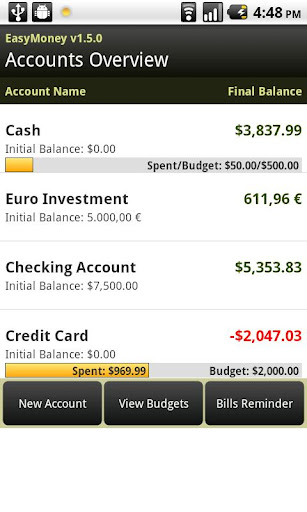
The man interface of program including three accounts default (Cash, Credit Card and Bank Account) and three buttons:

* New Account: User can create new account with Account Name, Currency, Description, Starting Balance…
* View Budget: User can view all accounts that they created. In each account user can Add Transaction (Income or expense). In this function user can create new Budget. User also can use Bills Reminder in this function.
* Bills Reminder: Add new Reminder and Receive notifications of upcoming and overdue bills - for both one-time payments as well as recurring bills.

Interactive Reports & Graphs let you analyses income, expenses, cash flow and balance over various customizable date ranges.

Set monthly budgets on specific accounts and/or categories and monitor them via color-coded budget health bars charts

* **Advantages**
* Track income and expenses for multiple accounts in multiple currencies
* Support transfers between accounts
* Lock application using a 4-digit security passcode
* Backup data onto SD card
* Import & Export transaction data to .CSV & .QIF files
* **Disadvantages**
* Don’t have borrowing and lending function.
* To use this application, you must buy it.



*Figure 3: Easy Money Interface*

## Expected system

From above statistic about existing system, we have a table of comparison with functions that we have as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application Name** | **Money Lover** | **CWMoney** | **Easy Money** | **\*Money Tracker** |
| **Expense/Income** | Yes | Yes | Yes | Yes |
| **Schedule** | Yes | No | Yes | Yes |
| **Borrowing/Lending** | Yes | No | No | Yes |
| **Report/Chart** | Yes | Yes | Yes | Yes |
| **Warning** | Yes | Yes | Yes | Yes |
| **Synchronize Data** | No | Yes | Yes | Yes |
| **Input by NFC** | No | No | No | Yes |
| **Input by Q code** | No | No | No | Yes |
| **Export/Import** |  |  |  | Yes |
| **Score** |  |  |  | 9/9 |

Table 1: Comparative Matrix between the Existing System and the Money Tracker

Not only try to solve their some disadvantages, but also we will also learn from their good points. We expect that our system will bring to user all benefits and can be different with others.

There are some main features of our product:

* **Interface**: Our system will bring to user most friendly interface, easy to understand for user.
* **Function**: Our system also provides all function necessary to manage finance such as input expense, income, schedule, report, chart, warning and synchronize data. Furthermore, our system also helps user to input data faster by using NFC.
* Schedule: User can plan schedule follow week or month. In this function, user can input monthly budget or input total money for each category. System will compare expense/income with schedule, if it over budget, system will show warning message for user.
* Borrowing and lending: In this function, user can input interest to calculator the debts and loans.
* Expense and Income Management: helps user manage revenue and expenditure every month. In this function, user can input data faster by using NFC and Q code.
* Warning: reminder user when expenditures exceed plan or the maturity of borrowing or lending
* Report: helps user has an overview of the expenses in a month, compare with plan or other month.
* Synchronize: help storage data, when user change or lost mobile, all their data will not be lost and user can use this function to get back data and continue use application but don’t need input again.

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

As introduction in Report 1, name of this project is **Personal Finance Management**. Abbreviation **PFM**

### Problem Abstract

Our product helps people manage their personal finance in the way of the most quickly and usefully, without waste of time to remember and don't afraid forget any account.

### Project Overview

#### The current System

It will be a new system, developed by ourselves. There is no current system.

#### The Proposed System

There are some main functions of our product (when completed):

* Schedule: use to plan your future spending
* Revenue and expenditure management: helps user manage revenue and expenditure every month
* Borrowing and lending management: help user to easily manage borrowing and lending.
* Report: analyze your spending with reports and pie charts; help user has an overview of the expenses in a month, compare with plan or other month.
* Warning: alert you of upcoming bills or when expenditures exceed plan.
* Synchronize: help storage data

#### Boundaries of the System

* Final product is an application on Android
* Application for Vietnamese
* The system will have to work Android 2.3 or higher

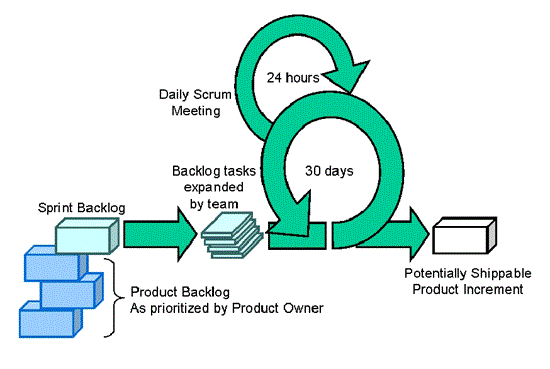
#### Development Environment

* OS:
* Developing language
* IDE:
* Database:

## Problem Organization

### Software Process Model

We apply new process model named: Scrum. All phases of project will follow this model.



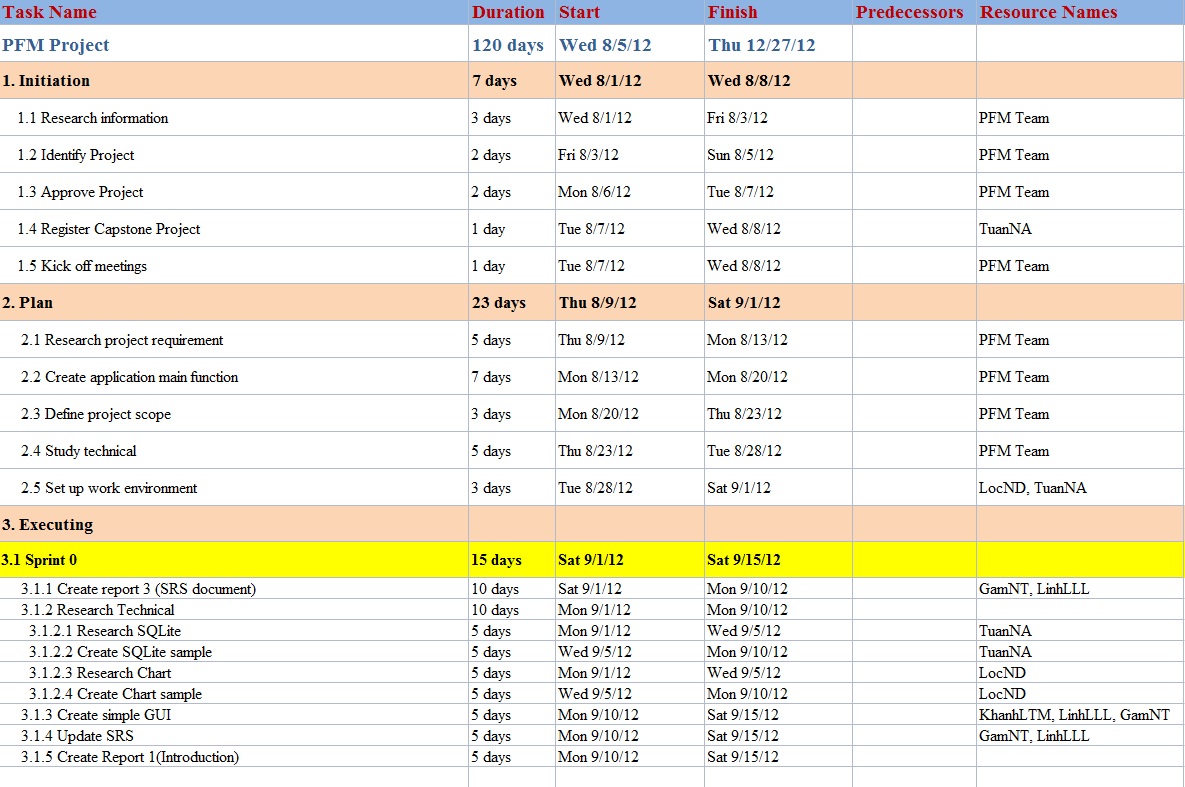
### Roles and Responsibilities

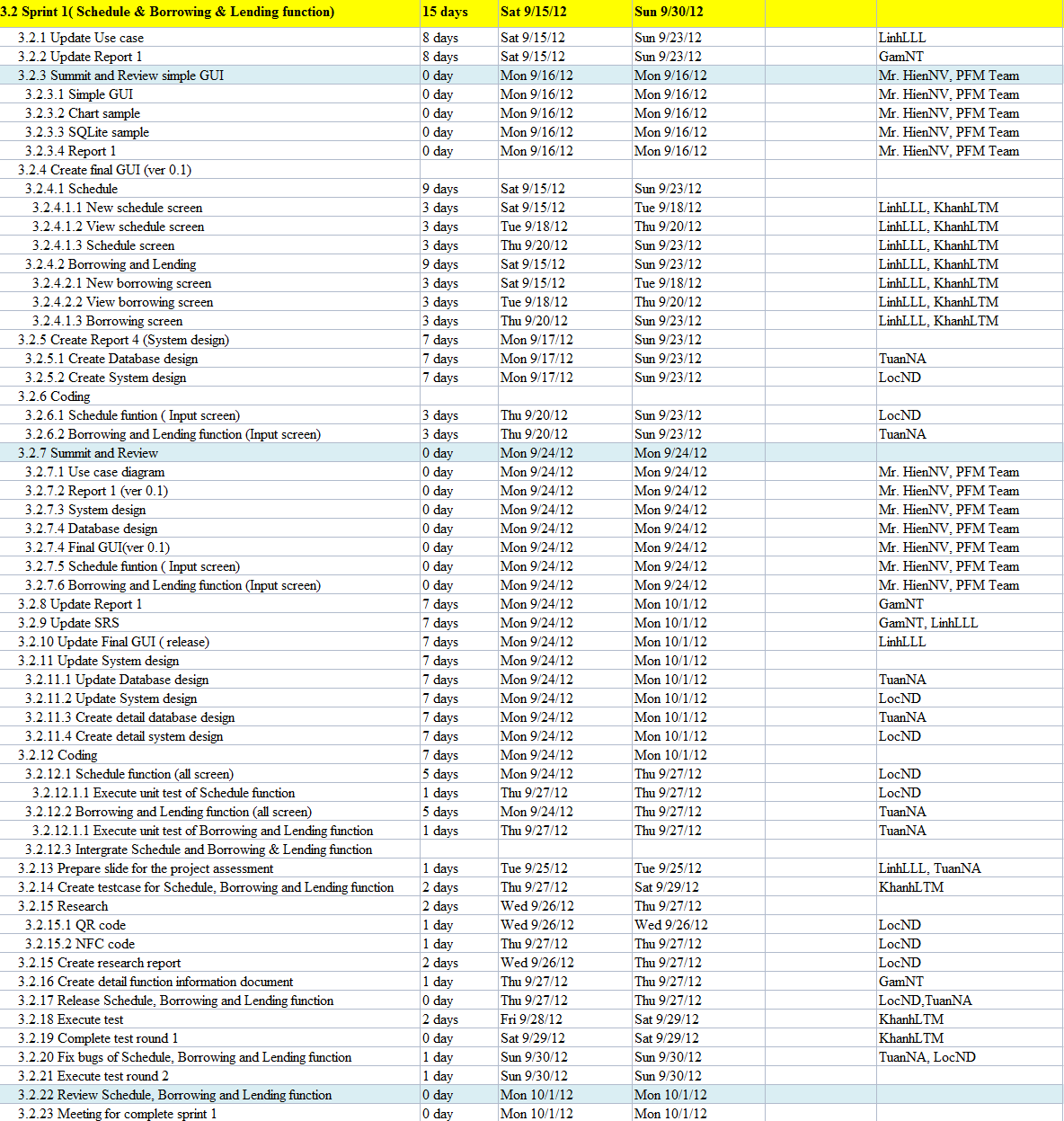
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Full Name | Role in Group | Responsibilities |
| 1 | Lai Le Le Linh | Team Leader  Designer  Tester | - Create project plan, take meeting, check status, and monitor activities...of team members.  - Promote the project implementation and evaluate steps of project.  - Write meeting minutes  - GUI design  - Testing  - Create use-case diagram |
| 2 | Nguyen Duc Loc | Technical Leader  Developer | - Support other members about technical issues.  - Build framework and guide for developers.  - Coding  - Unit testing  - Create System Design |
| 3 | Nguyen Anh Tuan | Developer | - Create Database Design  - Coding  - Unit testing |
| 4 | Le Thi Minh Khanh | Tester  Designer | - Create test case  - Testing  - Design GUI  - Write meeting minutes |
| 5 | Nguyen Thi Gam | Tester | - Create SRS  - Testing  - Create documents, reports  - Write meeting minutes |

### Tools and Techniques

## Project Management Plan

### Task Sheet: Assignments and Timetable



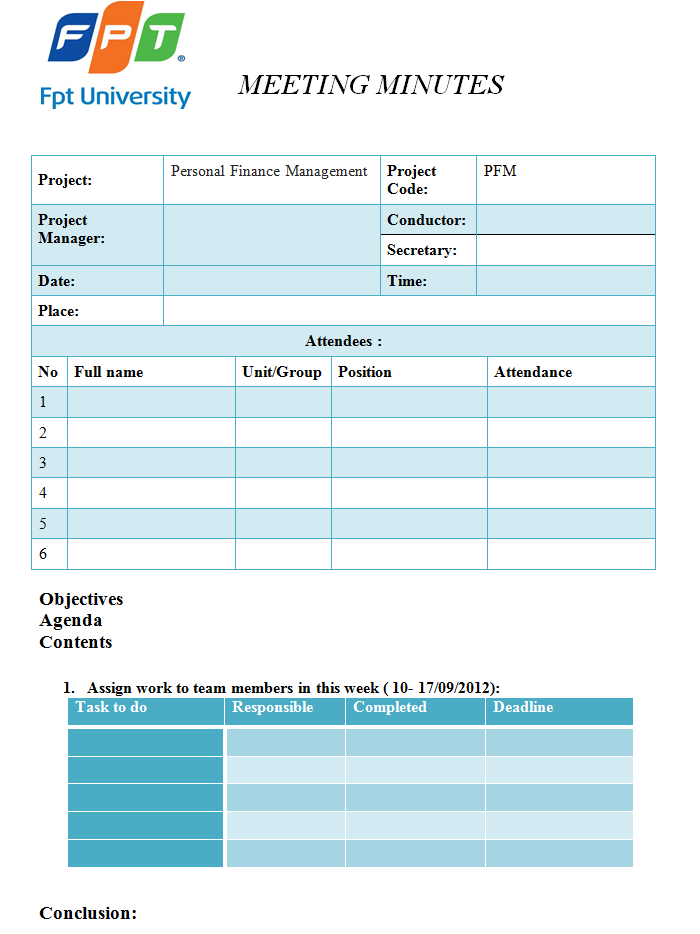






### All Meeting Minutes

Here is a template for meeting minute within the project’s life cycle.



## Coding Convention

We follow standard Java coding conventions. We add a few rules:

### Java Language Rules

#### Don't Ignore Exceptions

Sometimes it is tempting to write code that completely ignores an exception like this:

void setServerPort(String value) {

try {

serverPort = Integer.parseInt(value);

} catch (NumberFormatException e) { }

}

You must never do this. While you may think that your code will never encounter this error condition or that it is not important to handle it, ignoring exceptions like above creates mines in your code for someone else to trip over some day. You must handle every Exception in your code in some principled way. The specific handling varies depending on the case.

Acceptable alternatives (in order of preference) are:

* Throw the exception up to the caller of your method.

*void setServerPort(String value) throws NumberFormatException {*

*serverPort = Integer.parseInt(value);*

*}*

* Throw a new exception that's appropriate to your level of abstraction.

*void setServerPort(String value) throws ConfigurationException {*

*try {*

*serverPort = Integer.parseInt(value);*

*} catch (NumberFormatException e) {*

*throw new ConfigurationException("Port " + value + " is not valid.");*

*}*

*}*

* Handle the error gracefully and substitute an appropriate value in the catch {} block.

*/\*\* Set port. If value is not a valid number, 80 is substituted. \*/*

*void setServerPort(String value) {*

*try {*

*serverPort = Integer.parseInt(value);*

*} catch (NumberFormatException e) {*

*serverPort = 80; // default port for server*

*}*

*}*

* Catch the Exception and throw a new RuntimeException. This is dangerous: only do it if you are positive that if this error occurs, the appropriate thing to do is crash.

*/\*\* Set port. If value is not a valid number, die. \*/*

*void setServerPort(String value) {*

*try {*

*serverPort = Integer.parseInt(value);*

*} catch (NumberFormatException e) {*

*throw new RuntimeException("port " + value " is invalid, ", e);*

*}*

*}*

Note that the original exception is passed to the constructor for RuntimeException. If your code must compile under Java 1.3, you will need to omit the exception that is the cause.

* Last resort: if you are confident that actually ignoring the exception is appropriate then you may ignore it, but you must also comment why with a good reason:

*/\*\* If value is not a valid number, original port number is used. \*/*

*void setServerPort(String value) {*

*try {*

*serverPort = Integer.parseInt(value);*

*} catch (NumberFormatException e) {*

*// Method is documented to just ignore invalid user input.*

*// serverPort will just be unchanged.*

*}*

*}*

#### Fully Qualify Imports

When you want to use class Bar from package foo,there are two possible ways to import it:

import foo.\*;

Pros: Potentially reduces the number of import statements.

import foo.Bar;

Pros: Makes it obvious what classes are actually used. Makes code more readable for maintainers.

Decision: Use the latter for importing all Android code. An explicit exception is made for java standard libraries (java.util.\*, java.io.\*, etc.) and unit test code (junit.framework.\*)

### Java Style Rules

#### Use Javadoc Standard Comments

Every file should have a copyright statement at the top. Then a package statement and import statements should follow, each block separated by a blank line. And then there is the class or interface declaration. In the Javadoc comments, describe what the class or interface does.

*/\**

*\* Copyright (C) 2010 The Android Open Source Project*

*\**

*\* Licensed under the Apache License, Version 2.0 (the "License");*

*\* you may not use this file except in compliance with the License.*

*\* You may obtain a copy of the License at*

*\**

*\* http://www.apache.org/licenses/LICENSE-2.0*

*\**

*\* Unless required by applicable law or agreed to in writing, software*

*\* distributed under the License is distributed on an "AS IS" BASIS,*

*\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.*

*\* See the License for the specific language governing permissions and*

*\* Limitations under the License.*

*\*/*

*package com.android.internal.foo;*

*import android.os.Blah;*

*import android.view.Yada;*

*import java.sql.ResultSet;*

*import java.sql.SQLException;*

*/\*\**

*\* Does X and Y and provides an abstraction for Z.*

*\*/*

*public class Foo {*

*...*

*}*

Every class and nontrivial public method you write must contain a Javadoc comment with at least one sentence describing what the class or method does. This sentence should start with a 3rd person descriptive verb.

Examples:

*/\*\* Returns the correctly rounded positive square root of a double value. \*/*

*static double sqrt(double a) {*

*...*

*}*

Or

*/\*\**

*\* Constructs a new String by converting the specified array of*

*\* bytes using the platform's default character encoding.*

*\*/*

*public String(byte[] bytes) {*

*...*

*}*

#### Limit Variable Scope

The scope of local variables should be kept to a minimum (Effective Java Item 29). By doing so, you increase the readability and maintainability of your code and reduce the likelihood of error. Each variable should be declared in the innermost block that encloses all uses of the variable.

Local variables should be declared at the point they are first used. Nearly every local variable declaration should contain an initializer. If you don't yet have enough information to initialize a variable sensibly, you should postpone the declaration until you do.

One exception to this rule concerns try-catch statements. If a variable is initialized with the return value of a method that throws a checked exception, it must be initialized inside a try block. If the value must be used outside of the try block, then it must be declared before the try block, where it cannot yet be sensibly initialized:

*// Instantiate class cl, which represents some sort of Set*

*Set s = null;*

*try {*

*s = (Set) cl.newInstance();*

*} catch(IllegalAccessException e) {*

*throw new IllegalArgumentException(cl + " not accessible");*

*} catch(InstantiationException e) {*

*throw new IllegalArgumentException(cl + " not instantiable");*

*}*

*// Exercise the set*

*s.addAll(Arrays.asList(args));*

But even this case can be avoided by encapsulating the try-catch block in a method:

*Set createSet(Class cl) {*

*// Instantiate class cl, which represents some sort of Set*

*try {*

*return (Set) cl.newInstance();*

*} catch(IllegalAccessException e) {*

*throw new IllegalArgumentException(cl + " not accessible");*

*} catch(InstantiationException e) {*

*throw new IllegalArgumentException(cl + " not instantiable");*

*}*

*}*

*...*

*// Exercise the set*

*Set s = createSet(cl);*

*s.addAll(Arrays.asList(args));*

Loop variables should be declared in the for statement itself unless there is a compelling reason to do otherwise:

*for (int i = 0; i n; i++) {*

*doSomething(i);*

*}*

and

*for (Iterator i = c.iterator(); i.hasNext(); ) {*

*doSomethingElse(i.next());*

*}*

#### Follow Field Naming Conventions

* Non-public, non-static field names start with m.
* Static field names start with s.
* Other fields start with a lower case letter.
* Public static final fields (constants) are ALL\_CAPS\_WITH\_UNDERSCORES.

For example:

*public class MyClass {*

*public static final int SOME\_CONSTANT = 42;*

*public int publicField;*

*private static MyClass sSingleton;*

*int mPackagePrivate;*

*private int mPrivate;*

*protected int mProtected;*

*}*

#### Use Standard Java Annotations

Annotations should precede other modifiers for the same language element. Simple marker annotations (e.g. @Override) can be listed on the same line with the language element. If there are multiple annotations, or parameterized annotations, they should each be listed one-per-line in alphabetical order.<

Android standard practices for the three predefined annotations in Java are:

* @Deprecated: The @Deprecated annotation must be used whenever the use of the annotated element is discouraged. If you use the @Deprecated annotation, you must also have a @deprecated Javadoc tag and it should name an alternate implementation. In addition, remember that a @Deprecated method is still supposed to work.

If you see old code that has a @deprecated Javadoc tag, please add the @Deprecated annotation.

* @Override: The @Override annotation must be used whenever a method overrides the declaration or implementation from a super-class.

For example, if you use the @inheritdocs Javadoc tag, and derive from a class (not an interface), you must also annotate that the method @Overrides the parent class's method.

* @SuppressWarnings: The @SuppressWarnings annotation should only be used under circumstances where it is impossible to eliminate a warning. If a warning passes this "impossible to eliminate" test, the @SuppressWarnings annotation must be used, so as to ensure that all warnings reflect actual problems in the code.

When a @SuppressWarnings annotation is necessary, it must be prefixed with a TODO comment that explains the "impossible to eliminate" condition. This will normally identify an offending class that has an awkward interface. For example:

*// TODO: The third-party class com.third.useful.Utility.rotate() needs generics*

*@SuppressWarnings("generic-cast")*

*List<String> blix = Utility.rotate(blax);*

When a @SuppressWarnings annotation is required, the code should be refactored to isolate the software elements where the annotation applies.

#### Treat Acronyms as Words

Treat acronyms and abbreviations as words in naming variables, methods, and classes. The names are much more readable:

|  |  |
| --- | --- |
| **Good** | **Bad** |
| XmlHttpRequest | XMLHTTPRequest |
| getCustomerId | getCustomerID |
| class Html | class HTML |
| String url | String URL |
| long id | long ID |

Both the JDK and the Android code bases are very inconsistent with regards to acronyms, therefore, it is virtually impossible to be consistent with the code around you. Bite the bullet, and treat acronyms as words.

### Javatests Style Rules

#### Follow Test Method Naming Conventions

When naming test methods, you can use an underscore to seperate what is being tested from the specific case being tested. This style makes it easier to see exactly what cases are being tested.

For example:

*testMethod\_specificCase1 testMethod\_specificCase2*

*void testIsDistinguishable\_protanopia() {*

*ColorMatcher colorMatcher = new ColorMatcher(PROTANOPIA)*

*assertFalse(colorMatcher.isDistinguishable(Color.RED, Color.BLACK))*

*assertTrue(colorMatcher.isDistinguishable(Color.X, Color.Y))*

*}*

# Software Requirements Specifications

## Introduction

The introduction of the **Software Requirements Specification (SRS)** provides an overview of the entire **SRS**. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.

### Purpose

This SRS document describes functional and nonfunctional requirements for the Money Tracker, helps user manage personal finance. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system.

### Scope

The product that will be developed is a Money Tracker. This product uses both hardware, software, and many technologies to support the task at hand.

It is application for android and using NFC technology.

### Definitions, Acronyms, and Abbreviations

| Abbreviations | Description | Note |
| --- | --- | --- |
| PFM | Personal Finance Management |  |
| PM | Project Manager |  |
| NFC | Near Field Communication |  |
| TP | Test Plan |  |
| TC | Test Case |  |
| TR | Test Report |  |
| UT | Unit test |  |
| SRS | Software Requirements Specification |  |
| MSG | Message |  |

### References

### Overview

This document contains the complete description of the behavior of Personal Finance Management application and includes a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

## Specific Requirements

This section of the SRS contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the Use Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.

### External Interface Requirements

#### User Interfaces

* Application’s interfaces need to be clear and easy to use.
* Show message to confirm user in necessary cases.
* Display alert if user makes an error.
* Charts are clear and easy to understand.

#### Hardware Interfaces

* Google devices that support Android OS.
* NFC tags.
* Server.

#### Software Interfaces

* Java Development Kit (required for Android deployment).
* Android SDK (required for Android deployment).
* Eclipse (required for development environment).
* SQLite.
* Windows server.

#### Communications Interfaces

### Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

#### Revenue and expenditure management

This is the main and most important function of the system. It helps user manage revenue and expenditure every month. This function allows the user to input the income / expenses in every day and then save the data.

* Input expenses:
* The program will provide a default list of expenditures for user selected to input data (“Ăn uống”, “Điện”, “Nước”, “Sinh nhật”, “Tiệc tùng”, “Nhà”…)
* If expenses are not included in the list, the user can create a new account by selecting the "Khác". When select this option, immediately below will appear second textbox allows the user to manually enter other expenses. Then the program will automatically update the list to user do not have to re-enter when need enter a similar cost.
* After user saves one new expense in the day, at the bottom of the screen will display information:
* The total cost was used
* The remaining amount
* Use the QR code to input data
* Use NFC to input data
* Input revenue: Users can manipulate the same with expenses.

#### Schedule

This function helps user plan expenditure every month/week.

* The system will default time to start is the current date, in addition  
  the user also can manually enter start date.
* Next is total money user will expense in month/week (include total budget and detail budget)
* The user can enter a total budget, or just enter detailed budget.
* When the user does not enter a total budget, just enter the details budget,  
  system will automatically update the total budget.
* With detailed budget system will support a list of expenditures   
  (“Ăn uống”, “Điện”, “Nước”, “Sinh nhật”, “Tiệc tùng”, “Nhà”…) to user select. With one expenses not included in the list, the user can  
  select the "Khác" and create one new detail budget. And then system will save  
  and update it on the list available.
* If a user enters the detailed budget and total budget, when total of details budget greater total budget, the system will alert by displaying message with two options:
* Automatically update the total budget to match the detailed budget.
* Allow user update total budget or edit detail budget to accord.

#### Borrowing and lending management

This function allows the user to easily manage borrowing and lending.

With account payables or account receivables, the user will enter the necessary information such as:

* Total money (borrowing, lending or saving)
* Name (the lender, the borrower or where saving)
* Phone numbers and addresses.
* Term (borrowing, lending, saving).
* This type of interest rate.
* Interest (only support a fixed rate).
* In the case “Name” has in the contacts of mobile, the system will retrieve the information such as phone number and address of this person and user will not have to manually enter.
* User can view details borrowing, lending or saving, and interest received each day.
* When view, the system will sort accounts follow maturity closest.

#### Report

This function helps user has an overview of the expenses in a month, compare with plan or other month.

Beside detail report income and expense in the form of small reports, system also has some charts such as pie chart, growth charts to help user analyses and understand about their revenues and expenditures of the week, months and between the weeks and months.

#### Warning

This function uses to reminder users about expenditures, borrowing, lending and saving.

* When expenditures exceed plan.
* When the maturity of borrowing or lending.

#### Synchronize

This function will ensure the safety data for user, and support the users use multiple devices.  
Data created by user will be stored in the device, when a network connection, data will be synchronized with the server. And in other devices, if a network connection, this data  
will also be updated from the server.

### Use Cases

#### Use Case 1: Schedule

* Use-case Diagram

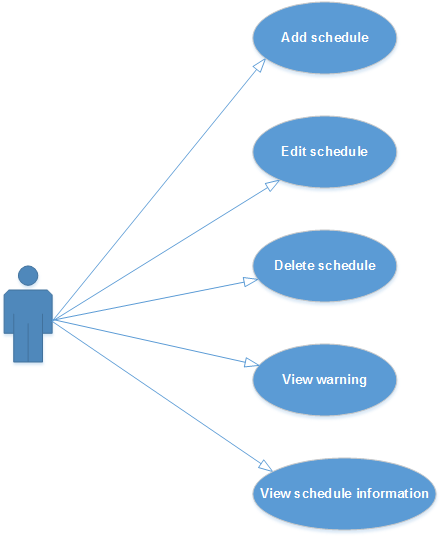


Figure 1: Schedule use-case diagram

* Use-case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- Create Schedule SPECIFICATION** | | | | |
| **Use-case No.** | UC SC.1 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Create Schedule** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can create budget, detail expense for current week or month.  **Goal:**  Allow user create budget and plan schedule.  **Triggers**  Touch schedule icon, second icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch create icon at the top- right concern of screen | Display create screen | | **3** | Touch “Tháng” toggle | Change toggle “Tháng” to “Tuần” | | **4** | Enter data into “Tổng ngân sách” field | Display entered data | | **5** | Touch save icon at the top- right concern of screen | * Return “Tháng” screen * Show message “Saved” * Display saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch create icon at the top- right concern of screen | Display create screen | | **3** | Enter data into all fields | Display all entered data | | **4** | Touch cancel icon beside save icon | * Return “Tháng” screen |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch create icon at the top- right concern of screen | Display create screen | | **3** | Enter data into “Ăn uống” field on screen | Display entered data | | **4** | Touch plus icon beside “Ăn uống” field | Display:   * Default value of combobox is: “Ăn uống” * One field | | **5** | Touch combobox “Ăn uống” | Display list options:   * Điện * Nước * Nhà * Sinh nhật * Khác | | **6** | Choose “Điện” option | Display combobox value is “Điện” | | **7** | Enter data into new field | Display entered data | | **8** | Touch save icon at the top- right concern of screen | * Return “Tháng” screen * Show message “Đã lưu” * Display saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch create icon at the top- right concern of screen | Display create screen | | **3** | Touch first date field | Choose desired time | | **4** | Enter data into “Tổng ngân sách” field | Display entered data | | **5** | Touch save icon | * Return “Tháng” screen * Show message “Đã lưu” * Display saved data |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch create icon at the top- right concern of screen | Display create screen | | **3** | Touch first date field | Choose desired time | | **4** | Enter data into “Tổng ngân sách” field | Display entered data | | **5** | Enter data into “Ăn uống” field on screen | Display entered data | | **6** | Touch plus icon beside “Ăn uống” field | Display:   * Default value of combobox is: “Ăn uống” * One field | | **7** | Enter data into new field | Display entered data | | **5** | Touch save icon | If total of detail budget greater than “Tổng ngân sách” display:   * Message: “Tổng chi phí đã vượt quá ngân sách, tăng thêm ngân sách?” * 2 buttons: “Có” and “Không” |     **Relationships:**  UC SC.2, UC SC.3, UC SC.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to create new schedule, user must accessed application | | **2** | In order to save schedule, user must enter data into mandatory field |   ***them_ke_hoach_vietsub*Screen:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- Edit Schedule SPECIFICATION** | | | | |
| **Use-case No.** | UC SC.2 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Edit Schedule** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can edit monthly or weekly budget, detail expense for user’s schedule which they created.  **Goal:**  Allow user edit budget and plan schedule.  **Triggers**  Touch “Kế hoạch” tab.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Kế hoạch” screen | | **2** | Choose one schedule in list schedule on screen | Display detail information of schedule | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Edit data in all fields | All data is changed | | **5** | Touch save icon at the top- right of screen | * Return “Tháng” screen * Show message “Đã lưu” * Display saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Kế hoạch” screen | | **2** | Choose one schedule in list schedule on screen | Display detail information of schedule | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Edit data in all fields | All data is changed | | **5** | Touch cancel icon beside save icon | * Return “Tháng” screen * No data is edited |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Kế hoạch” screen | | **2** | Choose one schedule in list schedule on screen | Display detail information of schedule | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Edit data in all fields | All data is changed | | **5** | Touch save icon at the top- right of screen | If total of detail budget greater than “Tổng ngân sách” display:   * Message: “Tổng chi phí đã vượt quá ngân sách, tăng thêm ngân sách?” * 2 buttons: “Có” and “Không” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch create icon at the top- right concern of screen | Display create screen | | **3** | Touch first date field | * Choose desired time * If desired time coincides with available time, show:   + Message: “Đã có kế hoạch trong thời gian này! Xin vui lòng chọn lại!” |   **Relationships:**  UC SC.1, UC SC.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to edit schedule, user must accessed application | | **2** | In order to edit schedule, “Kế hoạch” screen must have at least one schedule | | **3** | In order to save schedule, user must enter data into at least one field |   **Screen:**  **sua_ke_hoach_vietsub** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- Delete Schedule SPECIFICATION** | | | | |
| **Use-case No.** | UC SC.3 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Delete Schedule** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can delete schedule.  **Goal:**  Allow user delete schedule.  **Triggers**  Touch “Kế hoạch” icon, second icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen | | **2** | Long click on one schedule in list schedule | Display menu includes:   * “Chỉnh sửa” * “Xóa” | | **3** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **4** | Touch “Có” button | * Clear schedule * Show message: “Đã xóa” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Kế hoạch” icon, second icon at the bottom of screen | Display “Tháng” screen | | **2** | Long click on one schedule | Display menu includes:   * Chỉnh sửa * Xóa | | **3** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **4** | Touch “Không” button | * Return “Tháng” screen |   **Relationships:**  UC SC.1, UC SC.2, UC SC.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to delete schedule, user must accessed application | | **2** | In order to delete schedule, “Schedule” screen must have at least one schedule |   **Screen:**  **xoa_ke_hoach_vietsub** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- View Schedule SPECIFICATION** | | | | |
| **Use-case No.** | UC SC.4 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **View Schedule** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can view list of schedule or schedule detail.  **Goal:**  Allow user view schedule.  **Triggers**  Touch “Kế hoạch” icon, second icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen include list of monthly schedule   * Each schedule includes information below:   + Total budget  + Time  + Chart descripts this schedule | | **2** | Touch “Tuần” tab | Display “Tuần” screen include list of monthly schedule   * Each schedule includes information below:   + Total budget  + Time  + Chart descripts this schedule |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Kế hoạch” icon, second icon at the bottom of screen | Display “Tháng” screen | | **2** | Touch one schedule in list of schedule | Display “Chi tiết kế hoạch” screen includes all information below:   * Schedule time * List items of living cost * Amount of money is used for each item * Simple report |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen with content “Không có dữ liệu” | | **2** | Touch “Tuần” tab | Display “Tháng” screen with content “Không có dữ liệu” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen with content “Không có dữ liệu” | | **2** | Touch “Tuần” tab | Display “Tuần” screen include list of monthly schedule   * Each schedule includes information below:   + Total budget  + Time  + Chart descripts this schedule |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch second icon at the bottom of screen | Display “Tháng” screen include list of monthly schedule   * Each schedule includes information below:   + Total budget  + Time  + Chart descripts this schedule | | **2** | Touch “Tuần” tab | Display “Tuần” screen with content “Không có dữ liệu” |   **Relationships:**  UC SC.1, UC SC.2, UC SC.3  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to delete schedule, user must accessed application | | **2** | In order to delete schedule, “Kế hoạch” screen must have at least one schedule |   **Screen:**  chi_tiet_ke_hoach_vietsub | | | | |

#### Use Case 2: Income and Expenditure management

* Use-case Diagram

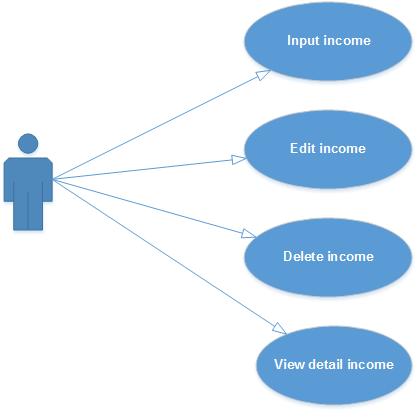


Figure 2: Income management use-case diagram

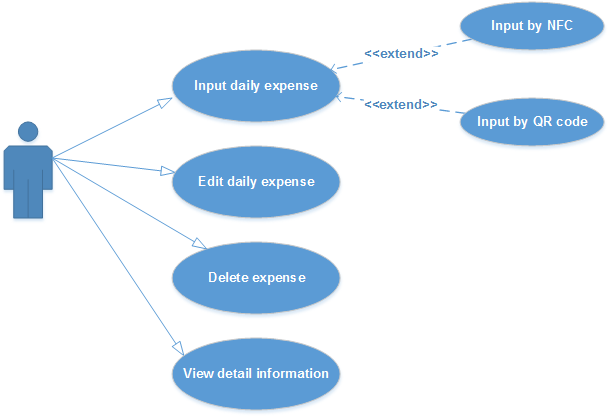


Figure 3: Expense management use-case diagram

* Use-case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- VIEW EXPENSE SPECIFICATION** | | | | |
| **Use-case No.** | UC MNG.1 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **View Expense** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 03/10/2012 | **Priority** | **High** | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Actor:**  User  **Summary:**  User can view monthly expense or daily expense.  **Goal:**  Allow user view detail expense.  **Triggers**  Touch management icon, first icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen includes:   * List of expense * Add new expense icon at the top-right of screen | | **2** | Touch one item of list | Display sub- item of selected item | | **3** | Touch one sub-item | Display “Chi phí ngày” screen |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen includes:   * List of expense * Add new expense icon at the top-right of screen | | **2** | Touch “Thu nhập” tab | Display “Thu nhập” screen |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen with content “Không có dữ liệu” | | **2** | Touch “Thu nhập” tab | Display “Thu nhập” screen with content “Không có dữ liệu” |   **Relationships:**  UC MNG.2, UC MNG.3, UC MNG.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to view expense, user must accessed application | | **2** | In order to view detail, user must touch one item of list expense |   **quan_ly_chi_tieu_view_tong_quanview_chi_tiet_chi_phiScreen:** | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- ADD DAILY EXPENSE OR INCOME SPECIFICATION** | | | | |
| **Use-case No.** | UC MNG.2 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Add Daily Expense or Income** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 03/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can input daily expense.  **Goal:**  Allow user input detail expense.  **Triggers**  Touch management icon, first icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Touch add new icon at the top- right concern of screen | Display “Thêm chi phí” screen | | **3** | Enter data into all fields | Display all entered data | | **4** | Touch save icon at the top- right concern of screen | * Return “Chi phí” screen * Show message “Đã lưu” * Display saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Touch add new icon at the top- right concern of screen | Display “Thêm chi phí” screen | | **3** | Enter data into all fields | Display all entered data | | **4** | Touch cancel icon besides save icon | * Return “Chi phí” screen * Nothing is saved |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Touch add new icon at the top- right concern of screen | Display “Thêm chi phí” screen | | **3** | Touch “Chi phí” toggle | Change “Chi phí” to “Thu nhập” | | **4** | Enter data into all fields | Display all entered data | | **5** | Touch save icon at the top- right concern of screen | * Return “Chi phí” screen * Show message “Đã lưu” * Display saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Touch add new icon at the top- right concern of screen | Display “Thêm chi phí” screen | | **3** | Touch first date field | Choose desired time | | **4** | Enter data into all remaining field | Display entered data | | **5** | Touch save icon | * Return “Chi phí” screen * Show message “Đã lưu” * Display saved data |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Touch create icon at the top- right concern of screen | Display “Thêm chi phí ” screen | | **3** | Touch first date field | Choose desired time | | **4** | Enter data into all field | Display entered data   * If amount of money make total expense greater than budget:   + show message: “Chi phí đã vượt quá ngân sách.” | | **5** | Touch save icon | * Return “Chi phí” screen * Show message “Đã lưu” * Display saved data |     **Relationships:**  UC MNG.1, UC MNG.3, UC MNG.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to add new daily expense, user must accessed application | | **2** | In order to save expense, user must enter data into mandatory field |   **them_moi_khoan_chiScreen:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- EDIT EXPENSE OR INCOME SPECIFICATION** | | | | |
| **Use-case No.** | UC MNG.3 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Edit expense or income** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 03/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can edit detail expense which they added.  **Goal:**  Allow user edit expense.  **Triggers**  Touch first icon at the bottom of screen  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen includes list of expense | | **2** | Choose one item of list expense on screen | Display view detail screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Edit data in all fields | All data is changed | | **5** | Touch save icon at the top- right of screen | * Return view detail screen * Show message “Đã lưu” * Display saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Choose one item of list expense on screen | Display view detail screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Edit data in all fields | All data is changed | | **5** | Touch cancel icon beside save icon | * Return view detail screen * No data is edited |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen includes list of expense | | **2** | Choose one item of list expense on screen | Display view detail screen includes detail information of expense | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | * Touch “Chi phí” toggle * Edit data in all fields | * Change “Chi phí” to “Thu nhập” toggle * All data is changed | | **5** | Touch save icon at the top- right of screen | * Return view detail screen * Show message “Đã lưu” * Display saved data |   **Relationships:**  UC MNG.1,UC MNG.2, UC MNG.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to edit expense/income, user must accessed application | | **2** | In order to edit expense/income, “Chi phí”/ “Thu nhập” screen must have at least one schedule | | **3** | In order to save what user edited, user must enter data into mandatory field |   **edit_khoan_chiScreen:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- DELETE EXPENSE OR INCOME SPECIFICATION** | | | | |
| **Use-case No.** | UC MNG.4 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Delete expense or income** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 03/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can delete expense or income.  **Goal:**  Allow user delete expense or income.  **Triggers**  Touch first icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Long click on one item in list expense | Display menu includes:   * “Chỉnh sửa” * “Xóa” | | **3** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **4** | Touch “Có” button | * Clear expense * Show message: “Đã xóa” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Schedule” tab | Display “Chi phí” screen | | **2** | Long click on one schedule | Display menu includes:   * Chỉnh sửa * Xóa | | **3** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **4** | Touch “Không” button | * Return “Chi phí” screen |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch first icon at the bottom of screen | Display “Chi phí” screen | | **2** | Touch “Thu nhập” tab | Display “Thu nhập” screen | | **3** | Long click on one item in list income | Display menu includes:   * “Chỉnh sửa” * “Xóa” | | **4** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **5** | Touch “Có” button | * Clear income * Show message: “Đã xóa” |   **Relationships:**  UC MNG.1, UC MNG.2, UC MNG.3  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to delete income/ expense, user must accessed application | | **2** | In order to delete income/expense, “Chi phí”/”Thu nhập” screen must have at least one schedule |   **Screen:** | | | | |

#### Use Case 3: Borrowing and Lending money management

* Use-case Diagram

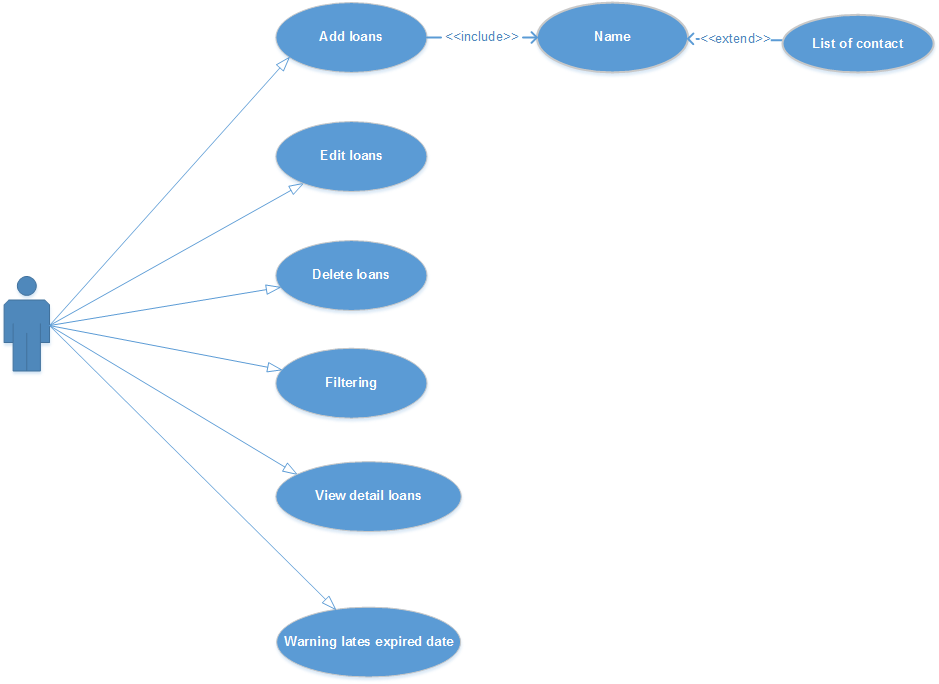


Figure 3: Borrowing and Lending money management use-case diagram

* Use-case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- Add New Loans SPECIFICATION** | | | | |
| **Use-case No.** | UC BL.1 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Add New Loans** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can add new loan.  **Goal:**  Allow user input detail information of loan.  **Triggers**  Touch “Vay nợ” icon, thirst icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Touch create icon at the top- right of screen | Display create screen | | **3** | Enter data into mandatory fields | Display all entered data   * If entered name is not available in mobile contact, display pop-up to add new contact | | **4** | Touch save icon at the top- right of screen | * Return “Vay” screen * Show message: “Đã lưu” * Display all saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Touch create icon at the top- right of screen | Display create screen | | **3** | Enter data into mandatory fields | Display all entered data | | **4** | Touch cancel icon besides save icon | * Return “Vay” screen * Nothing is added |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Touch create icon at the top- right of screen | Display create screen | | **3** | Touch “Vay” toggle | Change “Vay” to “Cho vay” | | **4** | Enter data into all fields | Display all entered data   * If entered name is available in mobile contact, system can be automatic display phone number and address | | **5** | Touch save icon at the top- right of screen | * Return “Vay” screen * Show message: “Đã lưu” * Display all saved data |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Touch create icon at the top- right of screen | Display create screen | | **3** | Touch “Vay” toggle | Change “Vay” to “Cho vay” | | **4** | Enter data into all fields except “Ngày đáo hạn” field | Display all entered data   * If entered name is available in mobile contact, system can be automatic display phone number and address | | **5** | Touch save icon at the top- right of screen | * Show message: “Bạn phải nhập ngày đáo hạn!” * Return create screen |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Touch create icon at the top- right of screen | Display create screen | | **3** | Touch “Vay” toggle | Change “Vay” to “Cho vay” | | **4** | Enter data into all fields except “Lãi suất” và “Ngày đáo hạn” field | Display all entered data   * If entered name is available in mobile contact, system can be automatic display phone number and address | | **5** | Touch save icon at the top- right of screen | * Return “Vay” screen * Show message: “Đã lưu” * Display all saved data |   **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to add borrowing money, user must accessed application | | **2** | In order to save data, user must enter data into mandatory field |   **Screen:**  ***khoan_vay_moi*** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- Edit Loans SPECIFICATION** | | | | |
| **Use-case No.** | UC BL.2 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Edit Loans** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can edit information of loan.  **Goal:**  Allow user edit detail information of loan.  **Triggers**  Touch “Vay nợ” icon, thirst icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Choose one borrowing money in list of borrowing money | Display detail information screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Change all of data in fields | Display all data is changed | | **5** | Touch save icon at the top- right of screen | * Return “Vay” screen * Show message: “Đã lưu” * Display all saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Choose one borrowing money in list of borrowing money | Display detail information screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Change all of data in fields | Display all data is changed | | **5** | Touch cancel icon besides save icon | * Return “Vay” screen * Nothing is changed |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Choose one borrowing money in list of borrowing money | Display detail information screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Touch “Vay” toggle | Change “Vay” to “Cho vay” | | **5** | Change data in all fields | Display all changed data | | **6** | Touch save icon at the top- right of screen | * Return “Vay” screen * Show message: “Đã lưu” * Display all saved data |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Choose one borrowing money in list of borrowing money | Display detail information screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **4** | Clear data in “Lãi suất” field | “Lãi suất” is empty field | | **5** | Clear data in “Ngày đáo hạn” | “Ngày đáo hạn” is empty field | | **6** | Touch save icon at the top- right of screen | * Return “Vay” screen * Show message: “Đã lưu” * Display all saved data |   **Exceptions:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Touch create icon at the top- right of screen | Display create screen | | **3** | Touch “Vay” toggle | Change “Vay” to “Cho vay” | | **4** | Clear data in one of field below:   * “Tên” * “Tổng số tiền” * “Ngày bắt đầu” * “Ngày đáo hạn” | Display all empty field | | **5** | Touch save icon at the top- right of screen | * Display message: “Xin hãy nhập Tên và Tổng số tiền! Xin cảm ơn!” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Choose one borrowing money in list of borrowing money | Display detail information screen | | **3** | Touch edit icon at the top- right of screen | Display edit screen | | **5** | Clear data in “Ngày đáo hạn” | “Ngày đáo hạn” is empty field | | **6** | Touch save icon at the top- right of screen | * Show message: “Hãy nhập ngày đáo hạn!” |   **Relationships:**  UC BL.1, UC BL.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to add borrowing money, user must accessed application | | **2** | In order to save data, user must enter data into mandatory field |   **Screen:**  **sua_khoan_vay_vietsub** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- Delete Loans SPECIFICATION** | | | | |
| **Use-case No.** | UC BL.2 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Delete Loans** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can delete loan.  **Goal:**  Allow user delete loan.  **Triggers**  Touch “Vay nợ” icon, thirst icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Long click on one borrowing in list borrowing | Display menu includes:   * “Chỉnh sửa” * “Xóa” | | **3** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **4** | Touch “Có” button | * Clear borrowing * Show message: “Đã xóa” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch “Vay nợ” icon, thirst icon at the bottom of screen | Display “Vay” tab includes list of borrowing money | | **2** | Long click on one borrowing in list borrowing | Display menu includes:   * “Chỉnh sửa” * “Xóa” | | **3** | Touch “Xóa” | Show pop-up confirm includes:   * Message: “Xóa?” * 2 buttons: “Có” and “Không” | | **4** | Touch “Không” button | * Return “Vay” screen |   **Relationships:**  UC BL.1, UC BL.2  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to add borrowing money, user must accessed application | | **2** | In order to save data, user must enter data into mandatory field |   ***vay_va_cho_vay_vietsub*Screen:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- View and Warning loans SPECIFICATION** | | | | |
| **Use-case No.** | UC BL.4 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **View and Warning loans** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/09/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can view list of loans or loan detail.  **Goal:**  Allow user view loan.  **Triggers**  Touch “Vay” icon, thirst icon at bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch thirst icon at bottom of screen | Display “Vay” screen includes list of information below:   * Each borrowing includes information below:   + “Tên”  + “Điện thoại”  + “Địa chỉ”  + “Kiểu lãi”  + “Lãi suất”  + “Ngày bắt đầu”  + “Ngày đáo hạn” | | **2** | Touch “Cho vay” tab | * Display “Cho vay” screen includes list of lending * Each lending includes information below:   + “Tên”  + “Điện thoại”  + “Địa chỉ”  + “Kiểu lãi”  + “Lãi suất”  + “Ngày bắt đầu”  + “Ngày đáo hạn” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch thirst icon at bottom of screen | Display “Vay” screen includes list of information below:   * Each borrowing includes information below:   + “Tên”  + “Điện thoại”  + “Địa chỉ”  + “Kiểu lãi”  + “Lãi suất”  + “Ngày bắt đầu”  + “Ngày đáo hạn” | | **2** | Touch “Cho vay” tab | * Display “Cho vay” screen with content “Không có dữ liệu” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch thirst icon at bottom of screen | Display “Vay” screen with content “Không có dữ liệu” | | **2** | Touch “Cho vay” tab | Display “Cho vay” screen includes list of information below:   * Each lending includes information below:   + “Tên”  + “Điện thoại”  + “Địa chỉ”  + “Kiểu lãi”  + “Lãi suất”  + “Ngày bắt đầu”   * + “Ngày đáo hạn” |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch thirst icon at bottom of screen | Display “Vay” screen with content “No data” | | **2** | Touch “Cho vay” tab | * Display “Cho vay” screen with content “Không có dữ liệu” |   **Relationships:**  UC BL.1, UC BL.2, UC BL.3  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to delete schedule, user must accessed application | | **2** | In order to delete schedule, “Schedule” screen must have at least one schedule |   **Screen:**  chi_tiet_ke_hoach_vietsub | | | | |

#### Use Case 4: Report

* Use-case Diagram

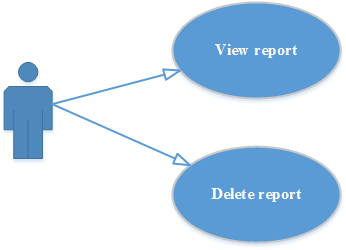


Figure 4: Report use-case diagram

* Use-case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- VIEW REPORT SPECIFICATION** | | | | |
| **Use-case No.** | UC RE.1 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **View report** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can view general weekly or monthly report. Besides, users can view detail report or chart which can compare income or expense between months.  **Goal:**  Allow user view report and compare income or expense between weeks/ months.  **Triggers**  Touch report icon, fourth icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fourth icon at the bottom of screen | Display “Báo cáo” screen default focus “Tháng” screen includes list of month which have information | | **2** | Touch one item of list | Default display “Chi tiết” screen of this item includes detail information | | **3** | Touch “Đồ thị” tab | Display “Đồ thị” screen includes pie chart which descripts income/expense of this item | | **4** | Touch compare icon on screen | Display list of months which have income or expense information | | **5** | Touch some items  Touch “Đồng ý” button | Display trend chart to compared between chosen months |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fourth icon at the bottom of screen | Display “Báo cáo” screen default focus “Tháng” screen includes list of months which have information | | **2** | Touch “Tuần” tab | Display “Tuần” screen includes list of weeks which have information | | **3** | Touch one item of list | Default display “Chi tiết” screen of this item includes detail information | | **4** | Touch “Đồ thị” tab | Display “Đồ thị” screen includes pie chart which descripts income/expense of this item | | **5** | Touch compare icon on screen | Display list of weeks which have income or expense information | | **6** | Touch some items  Touch “Đồng ý” button | Display trend chart to compared between chosen weeks |   **Relationships:** UC MNG.2, UC MNG.3, UC MNG.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to view report, user must have information of income or expense | | **2** | In order to compare between months/ weeks, user must have at least two months/ weeks |   **Screen:** | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- DELETE REPORT SPECIFICATION** | | | | |
| **Use-case No.** | UC RE.2 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Delete Report** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can delete weekly or monthly report.  **Goal:**  Allow user delete report.  **Triggers**  Touch report icon, fourth icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fourth icon at the bottom of screen | Display “Báo cáo” screen default focus “Tháng” screen includes list of month which have information | | **2** | Long click one item of list | Display dialog includes:   * Message: “Xóa?” * 2 buttons: “Có” và “Không” | | **3** | Touch “Có” button | * Clear this item * Show message: “Đã xóa!” | | **4** | Touch “Tuần” tab | Display list of weeks which have income or expense information | | **5** | Long click one item of list | Display dialog includes:   * Message: “Xóa?” * 2 buttons: “Có” và “Không” | | **6** | Touch “Có” button | * Clear this item * Show message: “Đã xóa!” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fourth icon at the bottom of screen | Display “Báo cáo” screen default focus “Tháng” screen includes list of month which have information | | **2** | Long click one item of list | Display dialog includes:   * Message: “Xóa?” * 2 buttons: “Có” và “Không” | | **3** | Touch “Không” button | * Return “Tháng” screen | | **4** | Touch “Tuần” tab | Display list of weeks which have income or expense information | | **5** | Long click one item of list | Display dialog includes:   * Message: “Xóa?” * 2 buttons: “Có” và “Không” | | **6** | Touch “Không” button | * Return “Tuần” screen |   **Relationships:** UC MNG.2, UC MNG.3, UC MNG.4  **Business Rules:**   |  |  | | --- | --- | | **No** | **Rule Description** | | **1** | In order to delete report, user must have information of income or expense |   **Screen:** | | | | |

#### Use Case 5: Warning

* Use-case Diagram

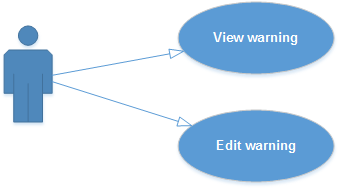


Figure 5: Warning use-case diagram

* Use-case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- WARNING SPECIFICATION** | | | | |
| **Use-case No.** | UC WAR | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Warning** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can change system warning for “Kế hoạch” and “ Vay và cho vay” functions  **Goal:**  Allow user change time, ring… of system warning  **Triggers**  Touch setting icon, fifth icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fifth icon at the bottom of screen. | Display “Thiết lập” screen includes list of item which user can change | | **2** | Change all item on screen | Display all changed data | | **3** | Touch “Đồng ý” button | Save all system change |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fifth icon at the bottom of screen. | Display “Thiết lập” screen includes list of item which user can change | | **2** | Touch first combobox on screen | Display list of choices for” Cảnh báo khi đến” label:   * 25% * 50% * 70% * 100% * Other | | **3** | Touch at least one item of list | Display chosen item | | **4** | Touch second combobox on screen | Display list of choices for ”Nhạc Chuông” label:   * Không có * Mặc định * Thư mục | | **5** | Touch “Thư mục” item | Display all song on user mobile phone | | **6** | Choose at least one song of list | Display chosen song title on second combobox | | **7** | Touch third combobox on screen | Display list of choices for “Nhắc lại” label:   * Không bao giờ * 10 phút * 30 phút * 1 tiếng * Other | | **8** | Choose at least one item of list | Display chosen item | | **9** | Touch “Đồng ý” button | Save all change of “Kế hoạch” warning functions |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch fifth icon at the bottom of screen. | Display “Thiết lập” screen includes list of item which user can change | | **2** | Touch fourth combobox on screen | Display list of choices for” Cảnh báo trước khi đáo hạn” label:   * 1 tiếng * 10 tiếng * 1 ngày * 1 tuần * Other | | **3** | Touch at least one item of list | Display chosen item | | **4** | Touch fifth combobox on screen | Display list of choices for ”Nhạc Chuông” label:   * Không có * Mặc định * Thư mục | | **5** | Touch “Thư mục” item | Display all song on user mobile phone | | **6** | Choose at least one song of list | Display chosen song title on second combobox | | **7** | Touch sixth combobox on screen | Display list of choices for “Nhắc lại” label:   * Không bao giờ * 10 phút * 30 phút * 1 tiếng * Other | | **8** | Choose at least one item of list | Display chosen item | | **9** | Touch “Đồng ý” button | Save all change of “Vay và cho vay” warning functions |   **E:\PFM\documents\SPEC\Sprint2\GUI\Image\canh_bao.pngScreen:** | | | | |

#### Use Case 6: Synchronize

* Use-case Diagram

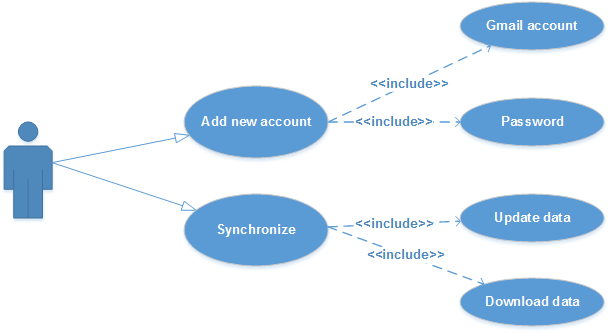


Figure 6: Synchronize use-case diagram

* Use-case Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE- SYNCHRONIZATION SPECIFICATION** | | | | |
| **Use-case No.** | UC SYN.1 | **Use-case Version** | | **1.0** |
| **Use-case Name** | **Synchronization** | | | |
| **Author** | **LinhLLL** | | | |
| **Date** | 08/10/2012 | **Priority** | **High** | |
| **Actor:**  User  **Summary:**  User can add new synchronize account.  **Goal:**  Allow user add new account for synchronize.  **Triggers**  Touch setting icon, last icon at the bottom of screen.  **Preconditions:**  Accessed to application  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch last icon at the bottom of screen | Display “Thiết lập” screen includes list of choices below:   * “Cảnh báo” * “Đồng bộ” | | **2** | Touch “Đồng bộ” | Display “Đồng bộ” screen includes “Thêm tài khoản” button | | **3** | Touch “Thêm tài khoản” button | Display gmail sign-in screen | | **4** | Enter data into all fields on screen | Display all entered data | | **5** | Touch sign-in button | * Return “Đồng bộ” screen * Display account on screen | | **6** | Touch automatic synchronize checkbox in “Thiết lập sơ bộ” section | * Display checked checkbox * Started the synchronization |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch last icon at the bottom of screen | Display “Thiết lập” screen includes list of choices below:   * “Cảnh báo” * “Đồng bộ” | | **2** | Touch “Đồng bộ” | Display “Đồng bộ” screen includes:   * Gmail account ( if any) * “Thêm tài khoản” button | | **3** | Touch automatic synchronize checkbox in “Thiết lập sơ bộ” section | * Display checked checkbox * Started the synchronization |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch last icon at the bottom of screen | Display “Thiết lập” screen includes list of choices below:   * “Cảnh báo” * “Đồng bộ” | | **2** | Touch “Đồng bộ” | Display “Đồng bộ” screen includes “Thêm tài khoản” button | | **3** | Touch “Thêm tài khoản” button | Display gmail sign-in screen | | **4** | Register new gmail account |  | | **5** | Sign-in with new account | * Return “Đồng bộ” screen * Display account * Started the synchronization |  |  |  |  | | --- | --- | --- | | **Step** | **Actor action** | **System response** | | **1** | Touch last icon at the bottom of screen | Display “Thiết lập” screen includes list of choices below:   * “Cảnh báo” * “Đồng bộ” | | **2** | Touch “Đồng bộ” | Display “Đồng bộ” screen includes:   * “Thiết lập sơ bộ” section   + Automatic synchronization checkbox   * “Tài khoản” section:   + All added account  + Default checked radio button beside latest synchronization account | | **3** | Touch radio button beside other account | Display checked radio button | | **4** | Touch automatic synchronize checkbox in “Thiết lập sơ bộ” section | Started the synchronization by checked account |   **Screen:** | | | | |

### Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level.

#### Performance

The main criteria for grading the performance of the system are:

* The verification process should be efficient and take less than 1 - 3 seconds to complete.
* System cans deployment at version different of android
* Content will display and done after less than 3 seconds.

#### Reliability

* System is capable to deal with unusually situation.
* The database has continuous back up and can be recovering if necessary.
* Not conflict with other software.

#### Availability

* Systems have to run 24/7
* System is capable to deal with unusually situation.
* The database has continuous back up and can be recovering if necessary

#### Security

To transfer data safely between client and server, encryption is used. The system shall authenticate all users with username and password. Users not authorized should not obtain access to the system.

#### Maintainability

The system shall have a documented source code to ease maintenance when this is needed. The system is designed based on separated modules, so it is easy to maintain or implement new version with many new functions.

#### Portability

The system will have to work android 2.3 or higher.

### Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.?

### Other Requirements

## Entity Relationship Diagram