UNIVERSITY OF Southampton

2015.11

COMP6226

Software Modeling and Design

Coursework

BillSplit application project document

Abstract

BillSplit application improve the efficiency of the bill management, at the same time, it enables multiple users to split different types bills simultaneously. BillSplit application is a more convenient and easier billing management system, there are three main aspects of the system, including backstage database establishment and maintenance, integration of background frame and the development of client-side server. The database establishment required data consistency, integrity and security, etc. Client-side server required to achieve the full-featured, easy to operate, easy to user and so on.

In this paper, we use MVC framework to developed IOS-side BillSplit application, design module of the system separately. For information module, it include add, query, update and other function. For payment module, we complete the payment method through bank interface which provided by bank and then generate bills for management. Back-end use SQL server database as the development tool, developed in a relatively short period of time to meet the needs of the feasibility of the system, while ensuring post-maintenance and increased function module.

Key words: MVC, IOS app, Bill Management, BillSplit

Acknowledgement

The authors would like to thank the lecturer of this class, A. Rezazadeh, for his guidance and support, as well as every member in the team for their struggle to solve every problem and challenge as possible as they can. And also the thanks must go to the other classmates' generously share they experience and ideas with us.

Contents

I. Problem statement/Introduction

1.1 Current issue

Managing and sharing expenses is a tedious and complicated issue in our normal life. People who are meeting the situation of sharing expenses such as traveling or having meals in groups, are facing the problem of calculating the exact count of money that everyone need to pay, and someone pay more while someone pay less as sometimes they unable to find out the small change, which may lead to the consequence of being in debit.

1.2 Goals

The aim of this project is to build up a mobile phone application, to make it easier, faster and more convenient to manage the bills, handling debts and share expenses on line by using their mobile phone with the others, providing the following facilities:

- The online payment interface will allow users who using this application pay their fees online easily and conveniently by binding their bank accounts.
- Users are able to set the numbers of the expense sharing group members and the total count that this expense sharing group need to pay. What's more, users have the choice to set the percentage of the total price that everybody in the group need to pay for a reason.
- Merchants also have the choice to set up the expense sharing group for the customers, furthermore, they are capable of giving a discount of the total price in some specific situations. And they can cancel the transaction after making a mistake.
- System owners/admins is defined to manage the application, the information of the merchants and the users, also the transactions.

1.3 Method

Once a user or a merchant input the total count of the price (/with discount) and the number of the members, the system will be distributed equally or be distributed by the proportion set by the user or the merchant.

1.4 Scope

Aspect like payments, scanning QR code and location recording are not handled by this system. However the system may share information with neighboring systems, such as payments system, camera and map systems.

II. Requirement

It will introduce three main BillSplit users. Functional and non-functional requirement will be discussed followed by project constrains/limitations and use case diagram.

2.1 Functional Requirements

This project will analyse functional requirements and for different stakeholders: users (see in Table 2.1), merchant (see in Table 2.2) and supervisor/owner (see in Table 2.3); each requirement is labelled clearly with name and definition. Priority for each functional requirement will also be discussed.

2.1.1 Users

ID	Name	Definition

FR-1	Registration	Non-registered users should go to BillSplit Application to do the initial registration if the BillSplit Application has enough capacity (users can also phone or email to the BillSplit Owner to check the capacity before registration); common users can register by providing username(could be email address), name, password, gender, day of birth, address, mobile number and e-mail through the Client-side application, and users will receive a confirmation email to confirm their registration after they submit application form.
FR-2		When non-registered users do the initial registration, they should choose a type of users. There are two types of users: merchants and common users. Merchants can register by providing username (could be email address), merchant-name, password, address, mobile number and e-mail through the Client-side application, and they need to submit business permits or provide relevant proof, the system verify these materials through industrial and Commercial Bureau interface.
FR-3	Viewing general information	Both registered and non-registered users can view the general information about the BillSplit Application through the homepage of this application, e.g. Introduction, guidance. Only registered users can make a payment, create group, etc.
FR-4		On the public part of the homepage of the application, new users can find guidance/direction on how to register with the application.
FR-5	Log-in	Registered users can login to the system by providing correct username (or email address) and password.
FR-6	View and Manage (update/ change) personal information	Registered users can view their personal information after they log into the application; Registered users can update their personal information after they log into the application; This could be changing personal details, e.g. Address and mobile number.
FR-7		Registered users can click “forgot password” if they do not remember their password. They can reset their password and will receive a verification email to confirm the alteration.
FR-8		Registered users can click “forgot username” if they do not remember their username. Username will be sent to their phone number by system.
FR-9		If registered users no longer use or cannot access their provided E-mail address, they should update their new e-mail address by providing userID to confirm the identity. the username could be the new E-mail address automatically by the system due to the registered users can use E-mail to log in.
FR-10	View and Manage(add and delete) personal bank account information	Registered users can add bank account, including card holder and card number, etc. They can use bank account to make a payment for all types of payment. System confirm these bank account information by sending verification to Bank through bank interface.
FR-11		Registered users can view their added bank account after log into the application and select a specific bank account to be the preferred payment method.
FR-12		If registered users no longer use their bank account or the

		bank account has been cancelled by the Bank for some reasons, users can delete the bank account and its information. The system will send a verification message to the user's mobile phone
FR-13	Make a Payment (for common users)	Registered users can select a specific bank account to make a payment. They can paid it individually or shared expense with a group, this application can calculate expenses or discounts for each person who in the same share-group.
FR-14		When registered users complete the payment, the system will save the bill automatically after finishing the payment, the bill will contain transaction detail, sharing group, date and location.
FR-15	Viewing transaction detail and bill management	Registered users can log into application and view their transaction history about different types of bills and payments, e.g. Long-term expenses, Short-term expenses and Single expenses. They can check all information about their bill, such as, date and amount.
FR-16		Registered users can manage the bills by setting different types for bills or payments, there are three main types: Long-term recurring expenses, Short-term recurring expenses and Single expenses. Users can set only a specific types for bills.
FR-17		Registered users can view their bills or payments through different ways of classification. According to different conditions, bills or payments could be classified by date, location, type and amount. Users also can view bills or payments in different orders, their bills can be ordered by date or amount.
FR-18	Join and create expense sharing group	When registered common users shared a Single expense (e.g. splitting a bill at dinner), they can join the expense sharing groups which is created by Merchant, when merchants create a group, it will set up a group number and then users search the group number to join the group. Merchant can select a type of payment method (on average and by percentage) for users to pay. The system can calculate expenses automatically for each person who in the group.
FR-19		When registered users shared a Short-term expense (e.g. Travel costs). An expense sharing group can be created by the person who paid the bills, it will set up a group number and then other users search the group number to join the group, the system can calculate the expenses for each person automatically. This payment means a transfer to initiator's account. The person who create the group can select payment method (on average and by percentage)to pay.
FR-20	Setting discounts (for merchant)	Registered merchants can set discounts or special offers for specific expense sharing group, those common users who joined the group can have this discount and the system will calculate expenses automatically for each person.
FR-21	Setting reminder (for common users)	Registered users can set Long-term recurring expenses (e.g. Rent, groceries) and Short-term recurring expenses (e.g. food). These expenses should be paid monthly so that users can setting reminder for these expenses, the system

		will send reminder to Registered users' email address or mobile phone.
FR-22	Manage (add/remove) friends	Registered users can add friend to their friend list, group initiator they can invite their friends who are in the friend list; registered users can delete friends by removing friends from friend lists.

Table 2.1 Functional requirement 1 (Users)

2.1.2 Supervisor/Owner

ID	Name	Requirement Definition
FR-23	Log-in	Supervisor/owner should login to the system by providing correct authorised admin through server-side application.
FR-24	Update application general information	Supervisor should check upgrade news, events or important information (e.g. Update information and error) for the application regularly via system.
FR-25	Add/remove categories	Supervisor/owner can add/remove categories via system. (E.g. Other types of users.)
FR-26	Manage information for users including common users and merchant	Supervisor/owner can view and update information of common users and merchant. When user cancelled their account, supervisor has the right to delete details about them; the system will delete user's account automatically when username is removed from categories.
FR-27	Manage bills of users	When a user is removed from database, the system will delete all its bill information from database.

Table 2.2 Functional requirement 2(Supervisor/Owner)

2.1.3 Priority for functional requirements

This section will discuss the priority of each functional requirement through three main aspects: complexity, necessity and priority (see in table 2.3). It is better to analysis the requirements priority in order to know which functional requirement are most important (highlight in red colour) and must be achieved during the limited project period.

Requirement ID	Complexity(1-5)	Necessity(1-5)	Priority(1-43)
FR-1	2	5	4
FR-2	3	5	4
FR-3	1	3	2
FR-4	1	4	4
FR-5	2	5	4
FR-6	3	3	2
FR-7	1	5	5
FR-8	1	5	5
FR-9	3	5	3
FR-10	3	5	4
FR-11	2	4	3
FR-12	3	4	3
FR-13	4	5	5
FR-14	3	5	5
FR-15	3	5	4
FR-16	4	5	4
FR-17	4	5	5
FR-18	4	5	5
FR-19	4	5	5

FR-20	3	3	3
FR-21	3	4	5
FR-22	2	3	3
FR-23	2	4	5
FR-24	1	3	3
FR-25	1	5	5
FR-26	1	5	5
FR-27	1	5	5

Table 2.3 Priority for functional requirements

2.2 Non-functional Requirements

Table 2.4 will introduce what the non-functional requirement for this system are, by giving clearly name and definition.

ID	Name	Requirements Definition
NF-1	Usability	The system should have an intuitive interface; all the tasks should be simple: ➢ Users can learn make a payment and create a group without any training and instruction. ➢ All users (common users and merchants) who using this application can understand each task within 15 minutes of training.
NF-2		The speed should not be slow down in any time (load within 5seconds), e.g. At least 50 users use system at the same time.
NF-3	Security	The application should cater for the security needs of the database and system, i.e.: ➢ Only supervisor/owner who has authorised admin can enter the system to view users' and merchants' information, or update application's information; stakeholder without admin access rights cannot enter the database. ➢ The admin will be changed every year to decrease the risk form hacker.
NF-4		The application need to protect users username and password, i.e.: ➢ The system will send automatic emails to remind users to renew their password every year. ➢ When users set up their password through this application, the system will remind them to create high-level password (more than 6 characters and with at least one lowercase, uppercase and number).
NF-5		The proposed solution should conform to W3C mark-up validation standards.
NF-6	Performance	The application should be performing functions and load pages in 5seconds.
NF-7	Compatibility	The application should be accessed from main types mobile operation system (IOS).
NF-8	Data Integrity	The database should be incorporated with the application, i.e. ➢ Information about application should be updated weekly ➢ New bank account and payment process should be updated immediately.

Table 2.4 Non-functional requirements

2.3 Use case diagram

The project figures use case diagrams for different actors (users-see in Figure 2.1 and owner/supervisor) through the BillSplit Application, the relationship between different actors and business use cases are shown in clearly explanations.

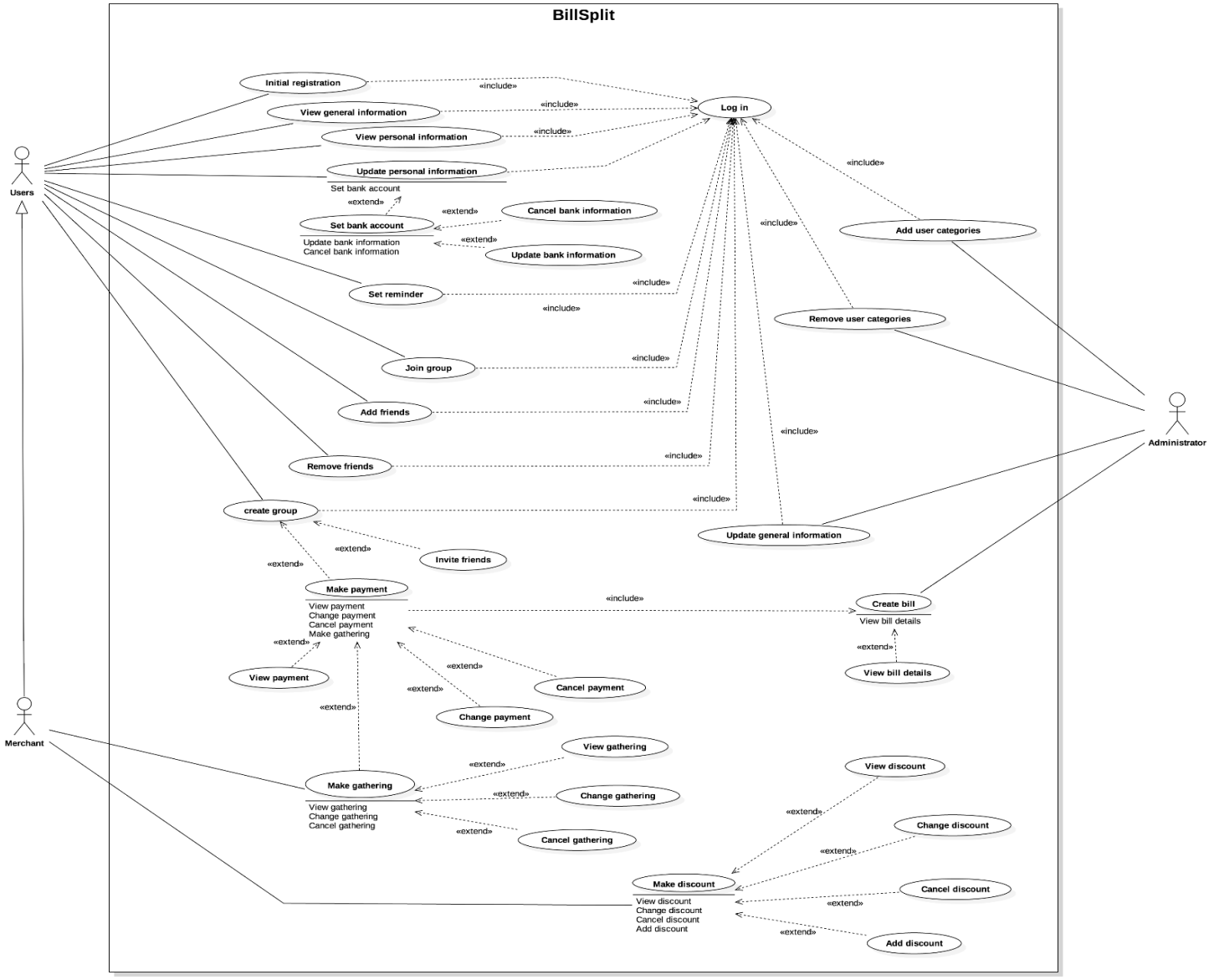


figure 2.1 BillSplit application Use Case Diagram

2.4 Constraints

2.4.1 Interface of the other systems

- **Payments system**

Users need to add (more than one) their bank accounts which they use to pay on the application.

- **Mobile phone's camera**

To join a new group, users have the choice to use the camera to scan the QR code that generated by the initiator.

- **Map system**

The system records every transaction location (not too accurate, but a probable area.) where the users pay their money.

- **Industrial & Commercial Bureau management system**

The first time that merchant, one of the roles in the system, sign up for a new account, they need to upload the proof of their identification which in keep with the data in the Industrial & Commercial Bureau management system.

2.4.2 Operation platform

Mobile phone only (IOS).

2.4.3 Operation environment

This application can only be used/processed under the environment of Internet connected, because the process of join groups, pay the money and add new friends, etc. are only can work when it has connected to the Internet.

2.4.4 Application language

For the reason of making this application popular all around the world, it has multiple language choices, so that users can change the system language in the setting as they want.

III. Design

3.1 Architecture

3.1.1 Overview

- **System Overview**

BillSplit Application is intended to help a user manage users' different types of expenses (i.e. Long-term recurring, Short-term recurring and Single expense), create bills in their account for management and share expenses with others. BillSplit is to provide the users the easier, faster and more convenient way to manage their bills than calculating and handing debts. Reminder about long-term recurring expense can be set to remind users to finish their payment on time.

- **System Context**

The system context is defined clearly in the SRS (Software Requirement Specification). Basically, the user is the main sink of the information. The user is also a major source of information and data. The other source of data is the bank from where user's bank account information is obtained.

- **Stakeholders of BillSplit**

The main stakeholder for the system are the individual users who might user the system, including common users and merchants, and the system designer/builder who will build BillSplit application. The main concerns of the two stakeholder are:

- For Users: The usability of the system and providing expenses sharing function, reminder function and bills management. Reasonable response time is also concern.
- For designer/builder: The system is easy to modify, particularly to handle future extensions mentioned in the SRS (i.e. the system is a multi-user system, which require use of database instead of files for keeping data.)

Hence, the key property for which the architecture is to be evaluated is the modifiability or expansibility of the system. Because the system is kind of payment system, response time performance is an essential factor for the system.

- **Scope of this Document**

In this document, we describe one specific architecture for BillSplit, discuss the advantage of various quality attributes of the application. We also provide the rationale for selecting the architecture. For architecture, we only consider the component view.

- **Definitions**

Name	Definition
Long-term recurring expenses	A bill that users need to pay regularly. This bills may be paid monthly or yearly, e.g. Rent, groceries and utilities.
Short-term recurring expenses	A bill that users need to pay regularly. This bills may be paid daily or weekly. E.g. Travel costs, food and hotel.
Single expenses	A bill that happened occasionally. This bill may be paid with friends or colleagues, e.g. Splitting a bill at dinner.
Expenses sharing	An event that split the amounts paid for goods and services among several group mates, this sharing is related to a series of payment operations.
Transaction	A real event that involves flow of personal money. In the context of expense, it is a payment to specific person which created the expenses sharing group.
Bill	A record that saved the details of the transaction and added the time when this transaction happened and location where this transaction happened.
Reminder	A message that remind users to pay their specific bills, e.g. Long-term recurring expense.

Table 3.1 System definitions

- **Acronyms**
  BS: BillSplit
  LRE: Long-term recurring expenses
  SRE: Short-term recurring expenses
  SE: Single expenses

3.1.2 Architecture design

- **Architecture: The MVC Model**

This Architecture have three basic components: Model, View and Controller (MVC [1]). Model: this component dealing with business processes/state, it accepted the requested data from View, and return the final results. This is the most important part of MVC. View: it represents the user interface for IOS application, with the complexity of the application process and become large, the interface becomes challenging. An application may have many different views, MVC design pattern for processing view is limited to the view on the data acquisition and processing, as well as the user's request. Controller: it can be understood to receive a request from client, matching the model and view together to complete the user's request. Control layer does not do any data processing. For example, a user click on a link, the control layer after receiving the request, it only passes the user's information to the model, and the model has responsibility to processing and choose to meet the requirements of view to user. Thus, a model may correspond to multiple views, a view may correspond a plurality of models.

The main function of BillSplit will be concluded in the Model layer and be triggered by the controller layer (e.g. Generate group) and the interface will be displayed through View layer.

BillSplit choose to use this model because of some advantages:

Firstly, the most important ability of MVC model it to have multiple views for a model correspondingly. In the current fast-changing needs of users, there may be a variety of ways to access the application requirement. This reduce the copied code, which reduce the amount of code maintain. It is obvious that this represent the low coupling, high reusability, maintainability, etc.[2] Secondly, because the returned data model without any display format, so these models can also be directly applied to the use of the interface. Thirdly, since an application is split into three levels, it is possible to change one of them and this can meet the requirement of the whole application. Changing business processes or business rules of an application will become simply

change the Model layer.

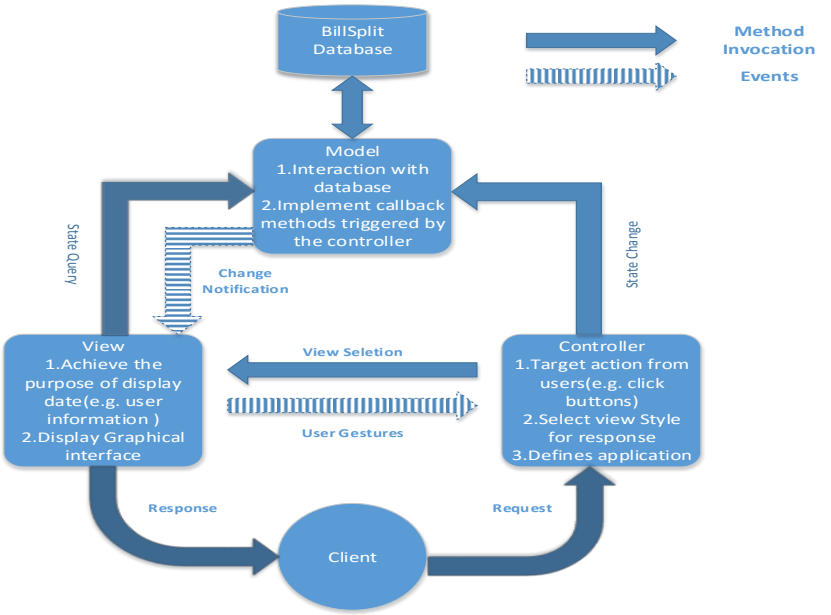


figure 3.1 The MVC Model

3.2 Detailed design

3.2.1 User interface design

Interface design tool: WireframeSketcher [3]

The application is an IOS [4] system APP, we obey some standards that the IOS APP ask for following, for example, every sub-page need a 'back' process and use natural icon that user can easily recognize and use it, etc. Also, the UI design obey the roles of making it Match between system and the real world, use language, wording, etc. that is consistent with the users' expectations, support rapid and easy learning of the system and support recognizing features and their associated actions, etc. The following pictures show the main functions of the BillSplit system.

As we can see user homepage in figure 3.2 (on the left), several function shortcuts are displaying on the screen. Four main ones (Scan, join group, new group, transaction) are on the top, so that user can easily and conveniently to access these function modules. Some definitions of the homepage are as follows:

name	definition
Scan	User can join a new expense sharing group by scanning the QR code generated by the initiator.
Join group	Except scanning QR code, user also can use 'Join group' to input the group number to join the group.
New group	To create a new expense sharing group.
Transaction	User can view his/her transactions history here.
Bank card	View and bind user's bank accounts.
Nearby	By using the interface of map system, use are able to view the nearby merchants.
Rent	To generate a house renting group (equally or percentage).
Travel	To generate a traveling group (equally or percentage).
Message	All the messages that user receive will be stored here, e.g. remind user to pay the house renting fees, traveling fees or being pulled into a new group and so on.
Support/help	There some guidance and introduction to lead user to use this

	application more easily.
Me	All the user's information (e.g. personal information, my group, bank account, etc.) can be viewed here.
Friends	A contact of all the friends the user has added.
	Share this application to the others through some popular social applications (e.g. WeChat, Mail, Facebook, Twitter, etc.).

Table 3.2 Homepage function module definitions

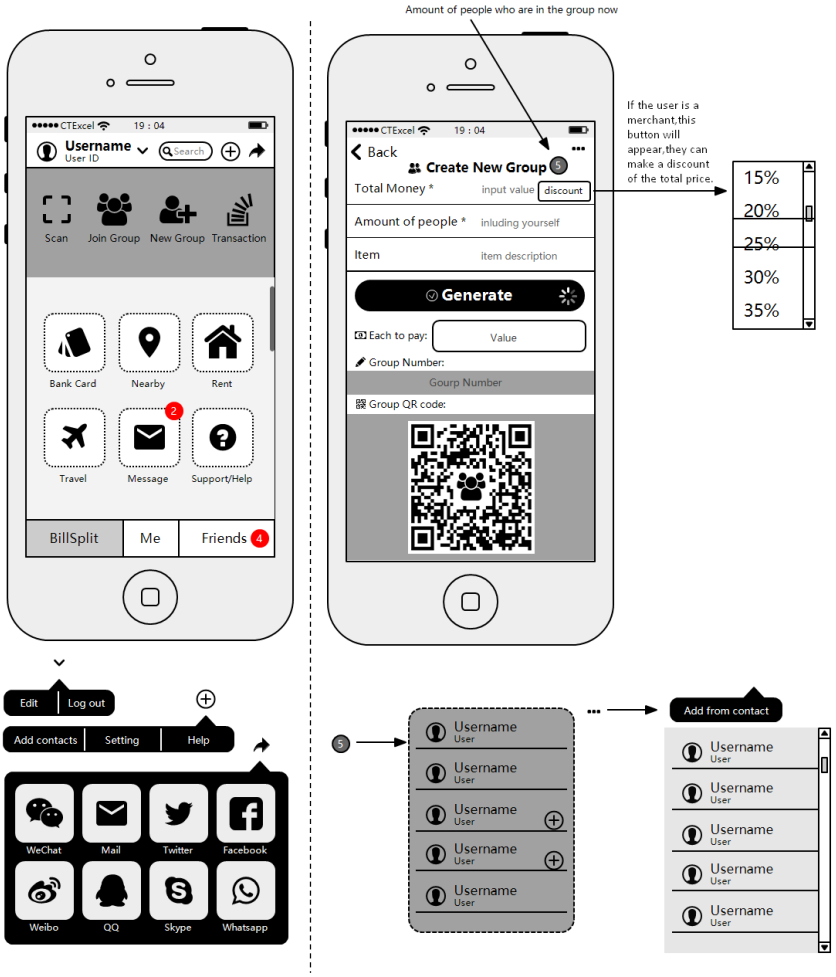


figure 3.2 User homepage (left) & generate new group (right) interface

On the right of figure 3.2 is the interface of creating new expense sharing group. After click the icon of 'New group' in the user homepage, this interface will appear. We assume that the price of the members in the group are equal division (go Dutch). Some definitions of this view are as follows:

name	definition
Total money	The total value of the price.
Amount of people	The total amount of the member to go Dutch.
Item	(Optional choice) the issue they pay for.
Group number	After the initiator create the group, a group number generates automatically.
Group QR code	After the initiator create the group, a group QR code generates automatically.
	By clicking this button, initiator can view the members who are in the group now.

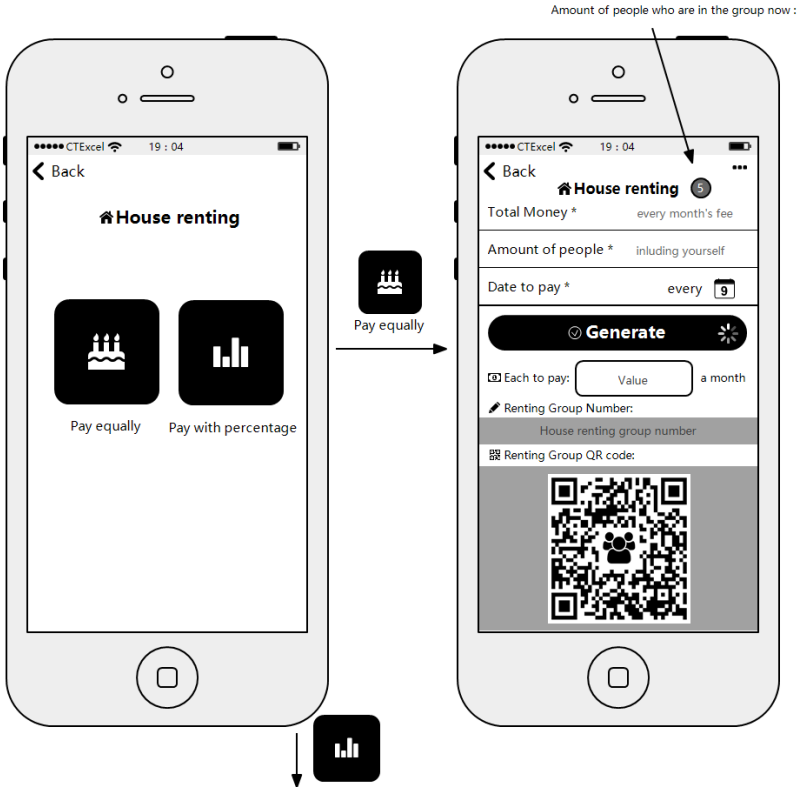
discount	This is an optional button, only appears when the user is a merchant. Merchants are able to give a discount of the total price in some special situations.
•••	Except let the others join the group by scanning the QR code or entre the group number, the initiator also can add the friends from contact straightly.

Table 3.3 Create new expense sharing group function module definitions

Figure 3.3 illustrates the house renting module BillSplit system. In the situation of renting a house or flat, user can set the payment date of every month. There are two options for users to choose (pay equally, pay with percentage). If the user choose pay equally, the new group interface is similar to generate new group interface in figure 3.1. Furthermore, if facing the case of everyone need to pay differently (bigger room pay more, smaller room pay less), user can choose the percentage that every member should pay. Some modules definitions are as follows:

name	definition
	Generate the group that every member in the group pay the same house renting fees.
	User can create a group that the percentage of every member should pay can be chosen.
**%	Every member's payment percentage can be view here.
9	Initiator can choose the date of every month which all the members should pay.

Table 3.4 Create new house renting fees sharing group function module definitions



(view nesxt page)

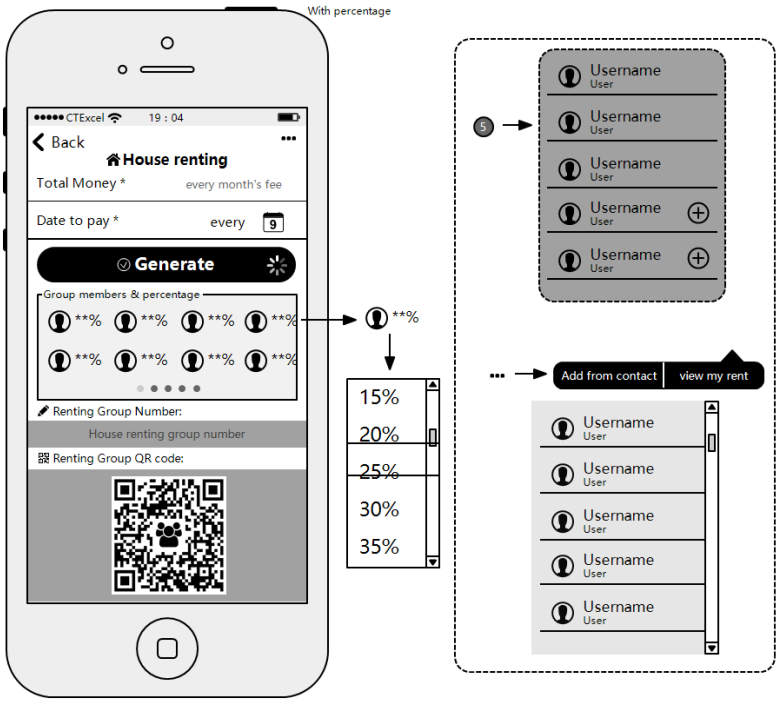


figure 3.3 House renting interface (equally & percentage)

3.2.2 Component design

- **User module**

User consists of two roles, common user and merchant. Merchant not only has some same functions as common user, like creating a new expense sharing group, viewing transaction history, binding the bank account and viewing the message, but also have a choice to make a discount of the total price when create a new expense sharing group for the other users (customers).

The common user also has some functions that merchant doesn't have, for example, split the fees of house renting or traveling in groups, and by scanning the QR code or input the group number to join any expense sharing group as they want.

- **Payment module**

When doing a payment, the system will deduct money from the bank account that the user has bound through the interface of the bank system. User are able to bind more than one bank account, so that they can have multiple choices when one account is lack of money.

- **Group module**

Group can divided into three types, common group, house renting group and traveling group. If a common group is generated, the fees will shared equally (every member in the group pay the same fees). A house renting group has two types, one is similar to the common group, and another one let the initiator to input the percentage that every member in the group should pay. Both of these two types can set the date of every month that all the member need to pay the rent fees. Travel type is something between the common one and the renting one, it can set the date as well, however not every month, just once, and it also can set the payment percentage as user want.

- **Bill/Transaction module**

In the transaction, user view all the bills generated before, which recorded the information about the group number, detail of the members, the count of the fees, the location where they pay the fees, also the time when the bill generated as well.

The relationships between the classes are clearly shown in figure 3.4. And in order to illustrate the process of the system's behavior, we use activity diagram to achieve this goal. (due to no more space to use, the activity diagram can be seen in appendices).

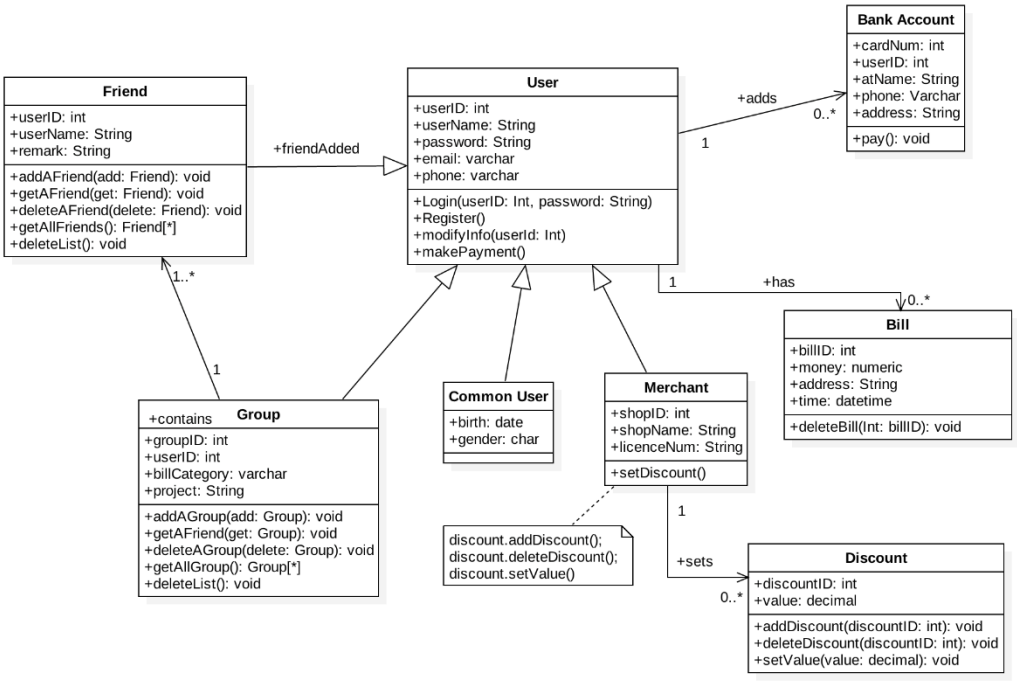


figure 3.4 BillSplit application class diagram

3.3 Database design

Database tool: SQL Server [5]

3.3.1 Summery sheet

Name	Function description
Common User	Describe user's personal information
Merchant	Describe shop's information
Group	Describe group's information
GroupMember	Show all the details about the members in the group
Reminder	Details about the message sent to the user
Payment	Connect the bank card and the user
Bill	Record the transaction history
Bank Account	Describe shop's information

3.3.2 Common User

Attributes	Type	Nullable	Note
UserID	INT(20)	NO	PK
UserName	STRING	NO	
Password	VARCHAR(50)	NO	
Email	VARCHAR(50)	NO	
Birth	DATE	YES	
Gender	CHAR(1)	YES	
Phone	VARCHAR(2)	YES	

3.3.3 Merchant

Attributes	Type	Nullable	Note
ShopID	INT(20)	NO	PK
ShopName	STRING	NO	
Password	VARCHAR(50)	NO	

Email	VARCHAR(50)	NO	
LicenseNum	VARCHAR(20)	YES	
Address	STRING	YES	
Phone	VARCHAR(2)	YES	

3.3.4 Friends

Attributes	Type	Nullable	Note
UserID	INT (20)	NO	PK, FK
Username	STRING	NO	
Remark	STRING	YES	

3.3.5 Group

Attributes	Type	Nullable	Note
GroupID	INT(20)	NO	PK
UserID	INT(20)	NO	FK
BillCategory	VARCHAR(50)	YES	
Project	VARCHAR	YES	

3.3.6 GroupMember

Attributes	Type	Nullable	Note
GroupID	INT(20)	NO	PK, FK
UserID	INT(20)	NO	PK, FK

3.3.7 Reminder

Attributes	Type	Nullable	Note
RMID	INT(20)	NO	PK
UserID	INT(20)	NO	FK
Content	TEXT(1000)	YES	
Time	DATETIME	NO	

3.3.8 Payment

Attributes	Type	Nullable	Note
UserID	INT(20)	NO	PK, FK
GroupID	INT(20)	NO	PK, FK
CardNum	INT(20)	NO	
UserName	STRING	NO	

3.3.9 Bill

Attributes	Type	Nullable	Note
BilID	INT(20)	NO	PK
UserID	INT(20)	NO	FK
GroupID	INT(20)	NO	FK
DiscountID	INT(20)	NO	FK
Money	NUMERIC	NO	
Time	DATETIME	NO	
Address	STRING	NO	

3.3.10 Discount

Attributes	Type	Nullable	Note
DiscountID	INT(20)	NO	PK

Value	DECIMAL	YES	

3.3.11 Bank Account

Attributes	Type	Nullable	Note
CardNum	INT(20)	NO	PK
UserID	INT(20)	NO	FK
ATname	STRING	NO	
Address	STRING	NO	
Phone	VARCHAR(2)	NO	

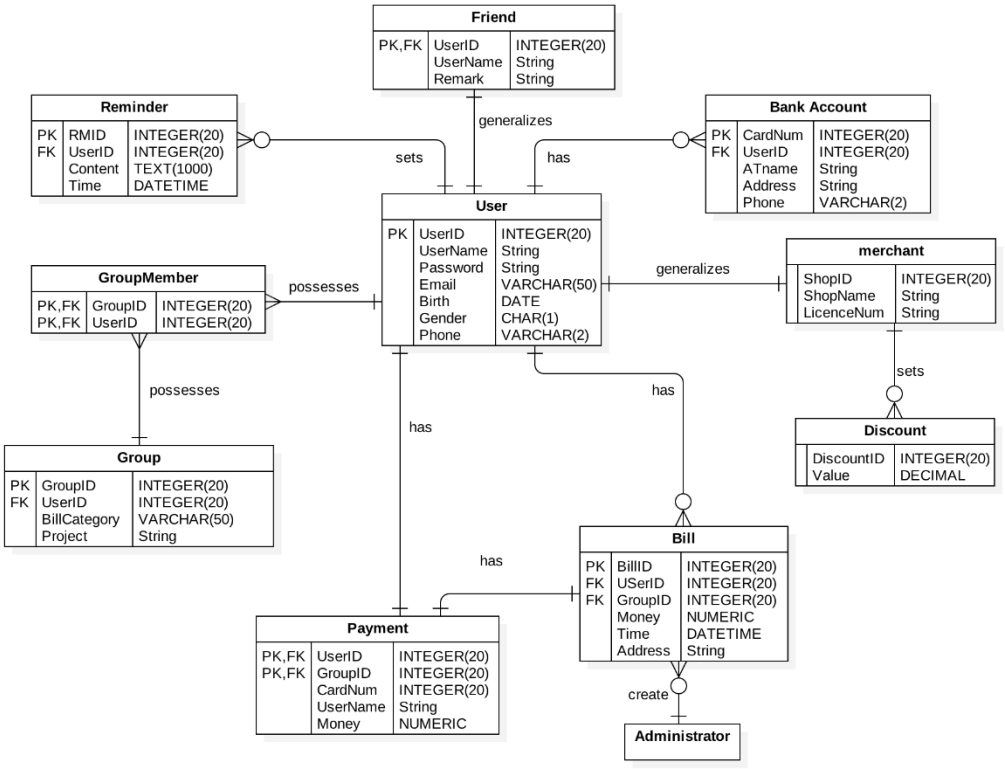


figure 3.5 BillSplit application EDR diagram

IV. Conclusion

BillSplit application is system that can effectively solve the problem about bill management and Splitting bills, the system enable manage different types of bills and payment, i.e. Long-term recurring expense, Short-term recurring expenses and Single expenses. The system enable filter different list of bills through setting specific conditions, bills can be sorted by different conditions (e.g. Date and location). This can make bill management faster, easier and convenient than manually calculating and handing debts.

In this paper, through analysis of user requirements, we determine the basic function of the system and finish detailed design which based on the requirements. The detailed design included user interface design, component design and database design. The system is developed on the IOS-side and used the MVC framework to achieve the basic function which included user information, bank information, groups and payments. For development of database, BillSplit application used SQL server to improve the usability of the data. At the same time, we described the Use Case Diagram, System Architecture Diagram, E-R Diagram, Class Diagram and activity Diagram of the system, this make the structure of the system become more intuitive and clear.

References

[1] Freeman, A 2013, Pro ASP.NET MVC 5, n.p.: [Berkeley, Calif.] : Apress, 2013., University of Southampton Library Catalogue.

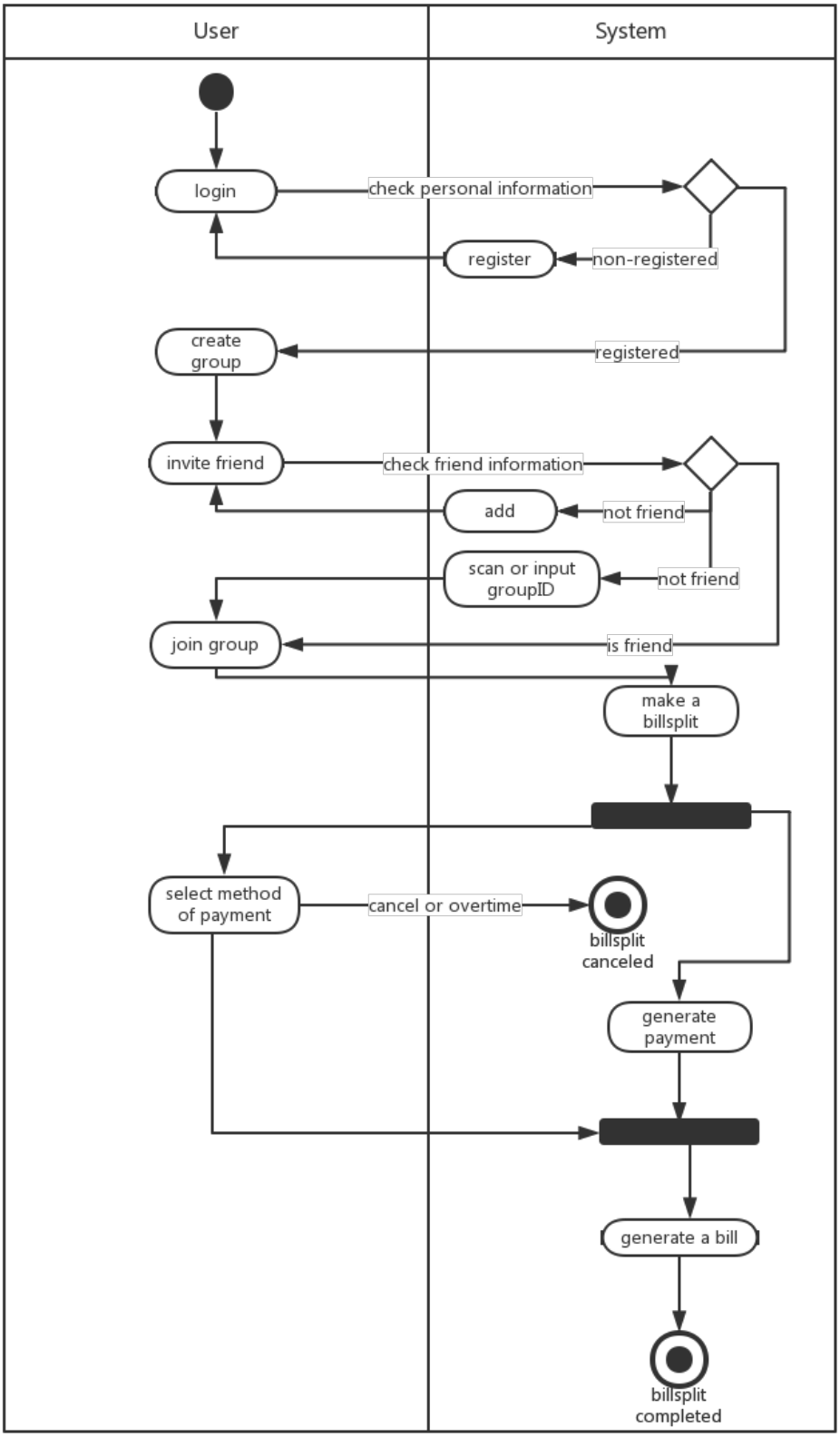
[2] Deck, P, Setiadi, M, Kurniawan, B, & Mayle, C 2014, Spring MVC : A Tutorial, n.p.: [Vancouver, Canada] : Brainy Software, 2014., University of Southampton Library Catalogue.

[3] Faranello, S 2012, Balsamiq Wireframes Quickstart Guide. [Electronic Resource], n.p.: Birmingham : Packt Publishing, 2012., University of Southampton Library Catalogue.

[4] Youcong, N, Bei, C, Peng, Y, & Chunyan, W 2014, 'A Framework for iOS Application Development', Journal Of Software (1796217X), 9, 2, pp. 398-403.

[5] Murugesan, M, Karthikeyan, K, & Sivakumar, K 2015, 'Analyzing integral components of SQL server databases', International Journal Of Applied Engineering Research, 10, 9, p. 24189-24200.

Appendices



BillSplit system activity diagram