Mobile Expense Management System

Group Members:

Haoyu Wang 104763720

Jinghua Long 104864306

Kerun Pan 104895029

Liangliang Xu 104760963

Rui Liu 104766936

Xiaoshuai Geng 104845608

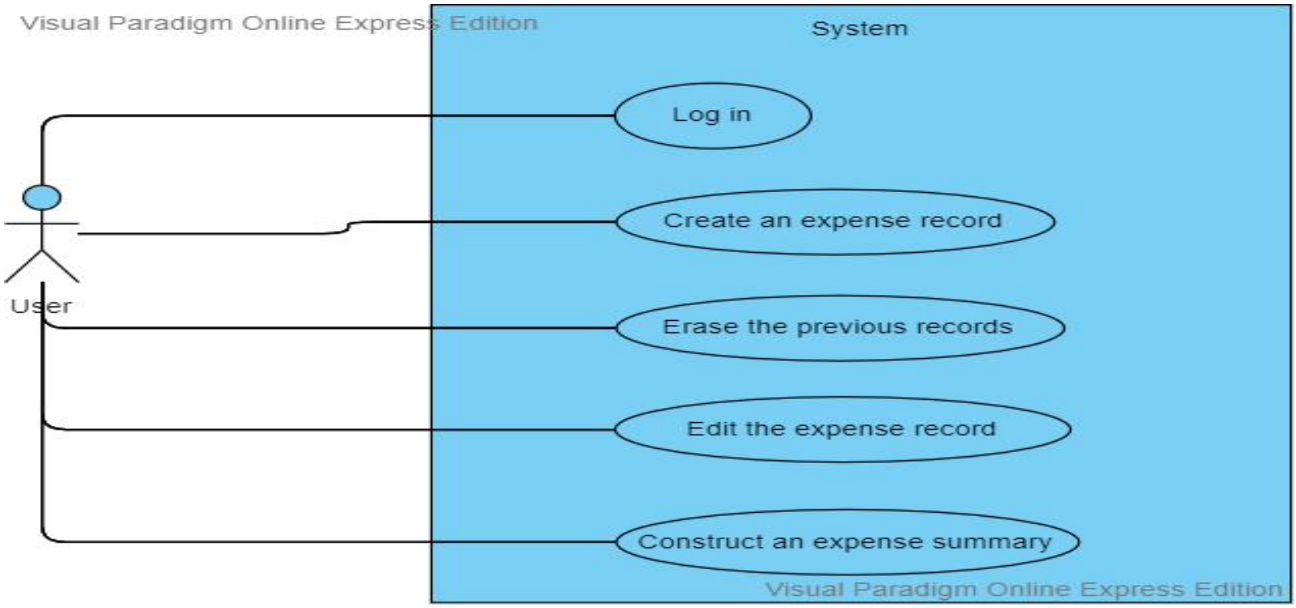
Zekun Zhang 104947128

Contents

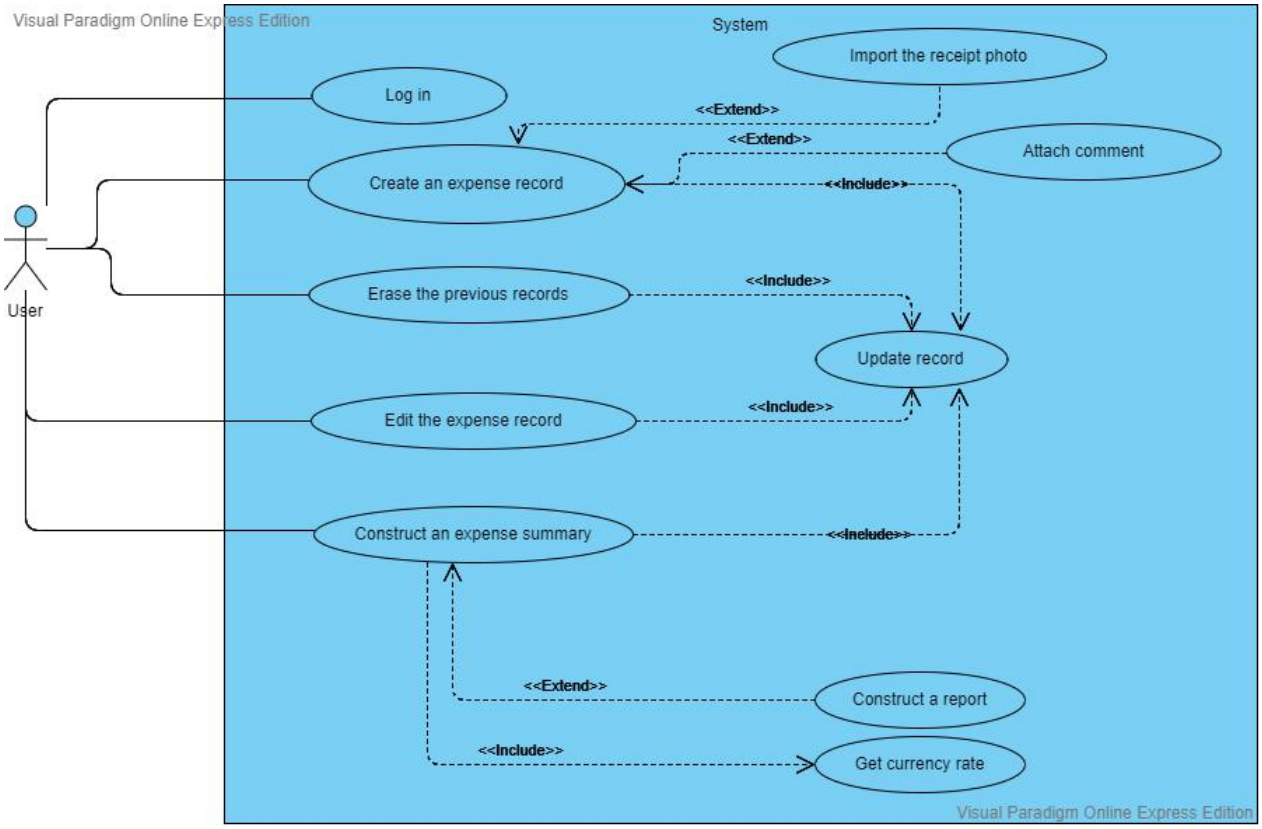
System Requirements:

ID	Requirements description	priority
RQ1	The system should allow user enter input of the detail of each transaction.	high
RQ2	The system should allow user to calculate the daily/monthly/yearly expense	high
RQ3	the system should be able to save user's entry.	high
RQ4	The system should allow user to abort an expense record.	high
RQ5	The system should allow user to erase the previous records.	high
RQ6	The system should allow user to choose what type of chart should be used for analyzing the expense data.	moderate
RQ7	The system should be able to construct an expense summary/report accommodate to user's requirement.	moderate
RQ8	The system should allow user to browse and re-enter the same entry from the user input history.	low
RQ9	Users can choose categories of the transactions, e.g. Travel, Food	moderate
RQ10	Users can import the receipt photo from phone gallery	moderate
RQ11	Users can enter the date of the expense or transaction	high
RQ12	Users can enter the amount of currency spent of the transaction	high
RQ13	Users can add description or comment on the expense entry	high
RQ14	Users can type the merchant name of the expense	high
RQ15	Users can select what kind of currency to use	high
RQ16	Users can a specific expense to the selected report	moderate
RQ17	The reports should calculate the total amount of expenses	high
RQ18	The reports should have the total amount of reimbursed values	high

Basic Use Case Model:



Structured Use Case Model:



User case description:

<table>
<tr><td colspan="2">Use Case Name: Create record</td><td>ID: UC-1</td><td>Priority: high</td></tr>
<tr><td colspan="4">Actor: Customer</td></tr>
<tr><td colspan="4">Description: This use case describes how the customer access the application to create a new record for one transaction</td></tr>
<tr><td colspan="4">Trigger: User want to add a new record and click the “add a new one” bottom to activate a event
Type: External</td></tr>
<tr><td colspan="3">Preconditions:
1. Customer is authenticated
2. Database is online</td><td></td></tr>
<tr><td colspan="3">Normal Course:
1. System display homepage
2. Customer clicks the “ add a new one ” bottom at homepage
3. System generate a new form to a list
4. System displays a pop-up window which contain a form
5. Customer enter details for each item inside the form</td><td>Information Steps:</td></tr>
<tr><td colspan="3">Alternative Courses:
1. Customer exits after saving the form:
1.1. Customer clicks the “save” bottom to updated and upload information to database
1.2. Customer close the pop-up window without prompts

2. Customer exits without saving:
2.1 Customer choose to close the pop-up window, a prompt pops up to ask if user want to save the form
2.2. System doesn’t upload this form to database if user choose “no”

3. Customer attach a comment to the form:
3.1. Customer clicks the “add comment” bottom inside the form
3.2. Customer enter words to a new pop-up window

4. Customer attach a photo to the form:
4.1. Customer clicks the “add photo” bottom inside the form
4.2. System displays two options: open folder explorer, open the camera</td><td></td></tr>
</table>

Postconditions: 1. One record is added to list 2. Customer return back to homepage after closing the form

Use Case Name: Login	ID: UC-2	Priority: high
Actor: Customer		
Description: This use case describes how the customer is authorized by the login operation using username and password		
Trigger: User requests login Type: External		
Preconditions: 1. System is connected to internet 2. Database is online		
Normal Course: 1. System display user interface, it prompts for username and password, Customer enters information 2. System validates the username and password, then display the home page	Information Steps:	
Alternative Courses: 1. User name is not valid: Retry getting username up to 3 times, then report failure 2. Password is not valid: Retry getting username up to 3 times, then report failure		
Postconditions: Customer is logged in successfully		

Use Case Name: Erase record	ID: UC-3	Priority: high
Actor: Customer		
Description: This use case describes how the customer access the application to delete an existing record for one transaction		
Trigger: User want to delete an existing record from the list and click the "delete record" bottom to activate an event Type: External		
Preconditions: 1. Customer is authenticated 2. Database is online		
Normal Course: 1. System display homepage 2. Customer clicks the "erase record" bottom at homepage 3. System displays a pop-up window which contain a list of records 4. Customer select a record; a prompt comes out to confirm the delete operation 5. System removes a record from the list, upload the whole list to database	Information Steps:	
Alternative Courses: " erase record " bottom is grey and unavailable if the list is empty or no items is selected		
Postconditions: 1. a record is removed from the list 2. list upload to database after each remove operation		

Use Case Name: Edit Record	ID: UC-4	Priority: high
Actor: Customer		
Description: This use case describes how the customer access the application to edit an existing record for one transaction		
Trigger: User want to edit an existing record from the list and click the "edit record" bottom to activate an event Type: External		
Preconditions: 3. Customer is authenticated 4. Database is online		
Normal Course: 1. System display homepage 2. Customer clicks the "edit record" bottom at homepage 3. System displays a pop-up window which contain a list of records 4. Customer select a record; a prompt comes out to confirm the edit operation 5. System display a pop-up window contains a form, which is the same as the form in the create record use case 6. Customer do edit on the form	Information Steps:	
Alternative Courses: 1. Customer exits after saving the form: 1.1 Customer clicks the "save" bottom to updated and upload information to database 1.2 Customer close the pop-up window without prompts 2. Customer exits without saving: 2.1 Customer choose to close the pop-up window, a prompt pops up to ask if user want to save the form 2.2. System doesn't upload this form to database if user choose "no"		
Postconditions: 3. One record in the list is modified 4. Customer return back to list after closing the form		

Use Case Name: Expense summary	ID: UC-5	Priority: high
Actor: Customer		
Description: This use case describes how the customer access the application to generate a daily/monthly/yearly expense summary		
Trigger: User want to summarize the expense and click the "Expense summary" bottom to activate an event Type: External		
Preconditions: 1. Customer is authenticated 2. Database is online		
Normal Course: 1. System display homepage 2. Customer clicks the "Expense" bottom at homepage 3. System retrieve a up to date record list from database, then display the record list 4. System retrieve currency rates (base on USD) for each tuple inside the list by accessing the database 5. Customer enter the time interval 6. System constructs a test report, in which the currency is represented as USD 7. Customer enter the report name	Information Steps:	
Alternative Courses: 1. Customer exits after saving the report: 1.1 Customer clicks the "save" bottom to save report as txt file in local folder 1.2 Customer close the pop-up window without prompts 2. Customer exits without saving: 2.1 Customer choose to close the pop-up window, a prompt pops up to ask if user want to save the report 2.2. System delete this report if user choose "no" 3. Customer select "daily/monthly/yearly" option instead of enter time interval 3.1 system summarize the group of records in the latest one day/month/year		
Postconditions: A report is saved in local folder as txt file		

Revised Functional Requirements

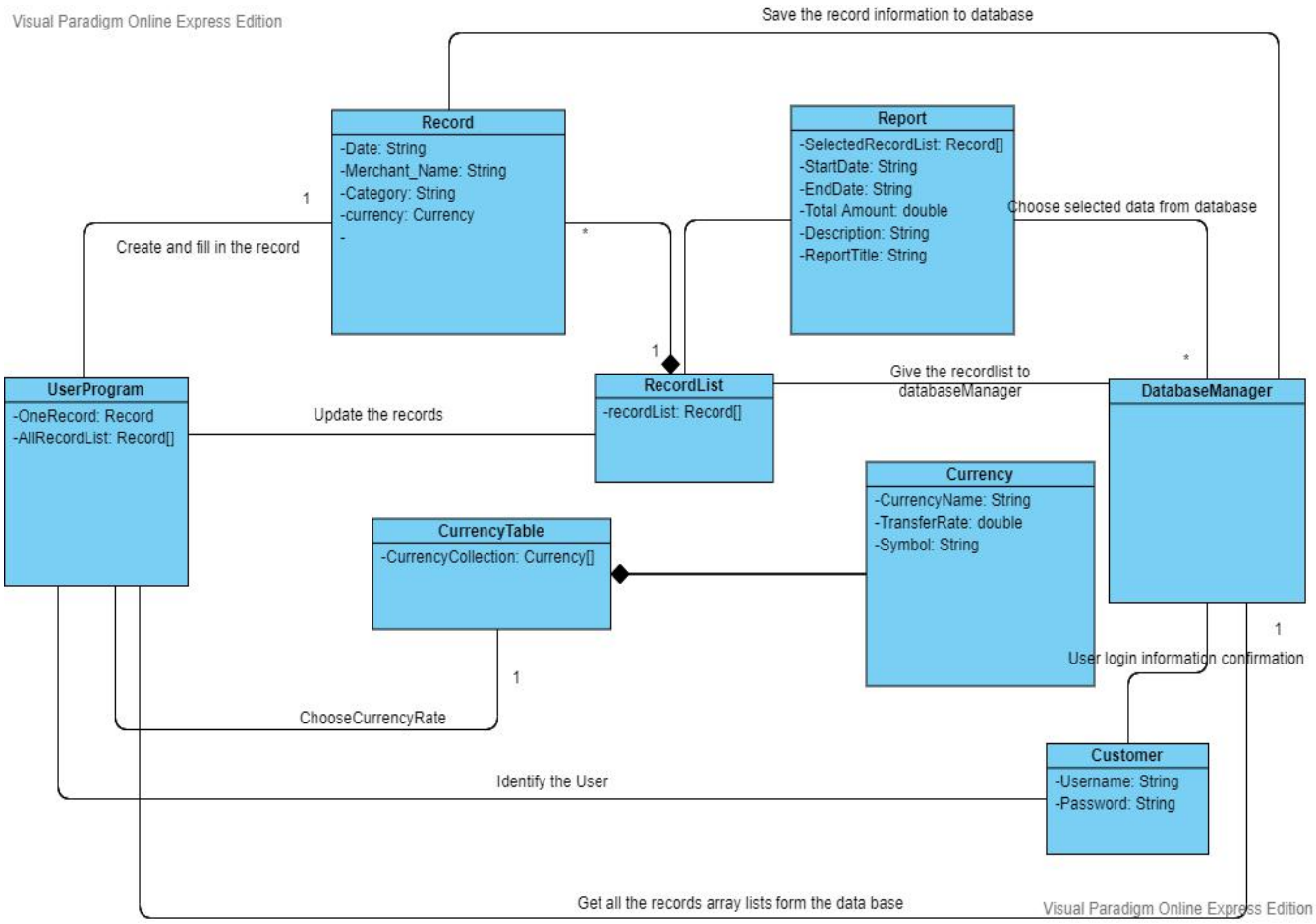
ID	Requirements description`	priority
RQ1	The system should allow user enter input of the detail of each transaction. Goal: create record Actor: customer	high
RQ2	The system should allow user to calculate the daily/monthly/yearly expense Goal: construct summary Actor:customer	high
RQ3	the system should be able to save user's entry. Goal: update record <-include—Save, create record, erase record, Import photo	high
RQ4	The system should allow user to abort an expense record.	high
RQ5	The system should allow user to erase the previous records. Goal: erase record Actor: customer	high
RQ6	The system should allow user to choose what type of chart should be used for analyzing the expense data. Goal: choose a chart <-extend—construct summary	moderate
RQ7	The system should be able to construct an expense summary/report accommodate to user's requirement. Goal: construct report <-extend—construct summary	moderate
RQ10	Users can import the receipt photo from phone gallery Goal: Import photo <-extend—create record Actor: customer	moderate
RQ13	Users can add description or comment on the expense entry Goal: attach comment <-extend— create record	high
RQ15	Users can select what kind of currency to use	high

	Goal: get currency rate <-include—construct summary (Need to access a currency rate table from database)	
RQ16	Users can a specific expense to the selected report	moderate
RQ19	User can modify the existing record Goal: edit record Actor: customer	high

Non-functional requirements

- **Operational:**
  - The software should be able to run properly on any Android devices
  - The software should be implemented in Java or Kotlin in Android platform
- **Performance:**
  - The system has a clear, simple-to-use graphic user interface
  - The system should only need internet connection when user submit their report
  - The system should work differently according to different types of users (employee / company)
- **Cultural / Political:**
  - The system should be able to switch into different currencies based on users' choice
- **Reliability:**
  - The system should have correct expense calculations
  - The system will save the modifying expense record as draft if user accidentally close the program or shun down the phone

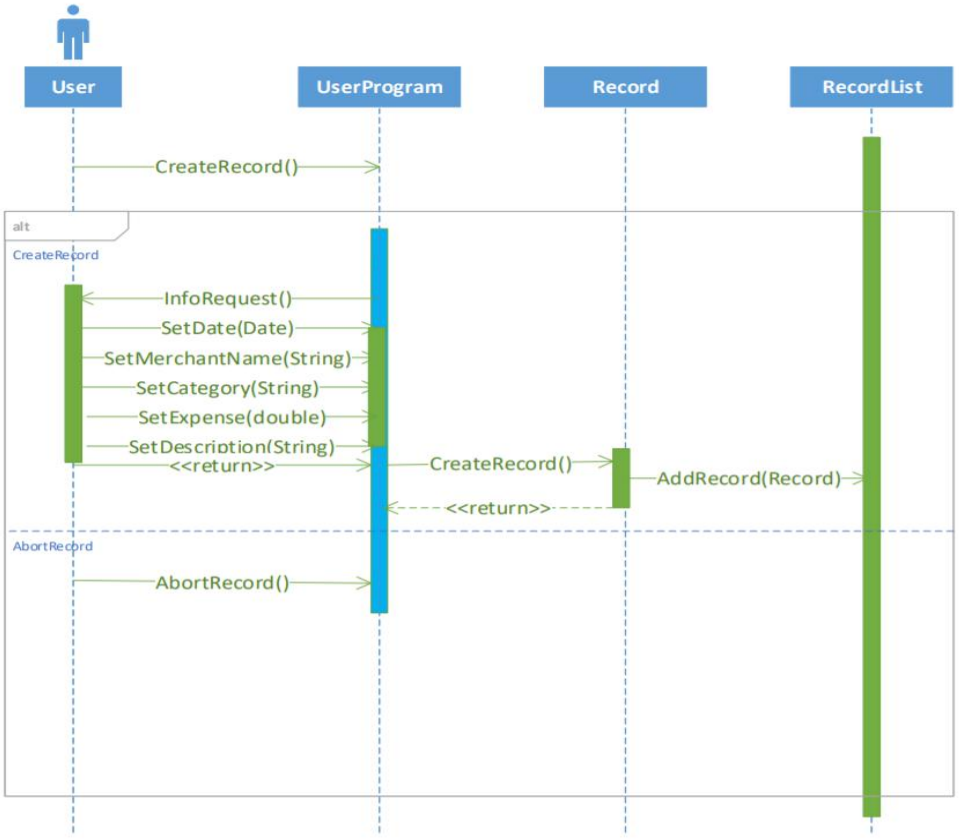
Domain Model:



This domain model display the basic relationship and interaction between classes in this program

Sequence Diagram:

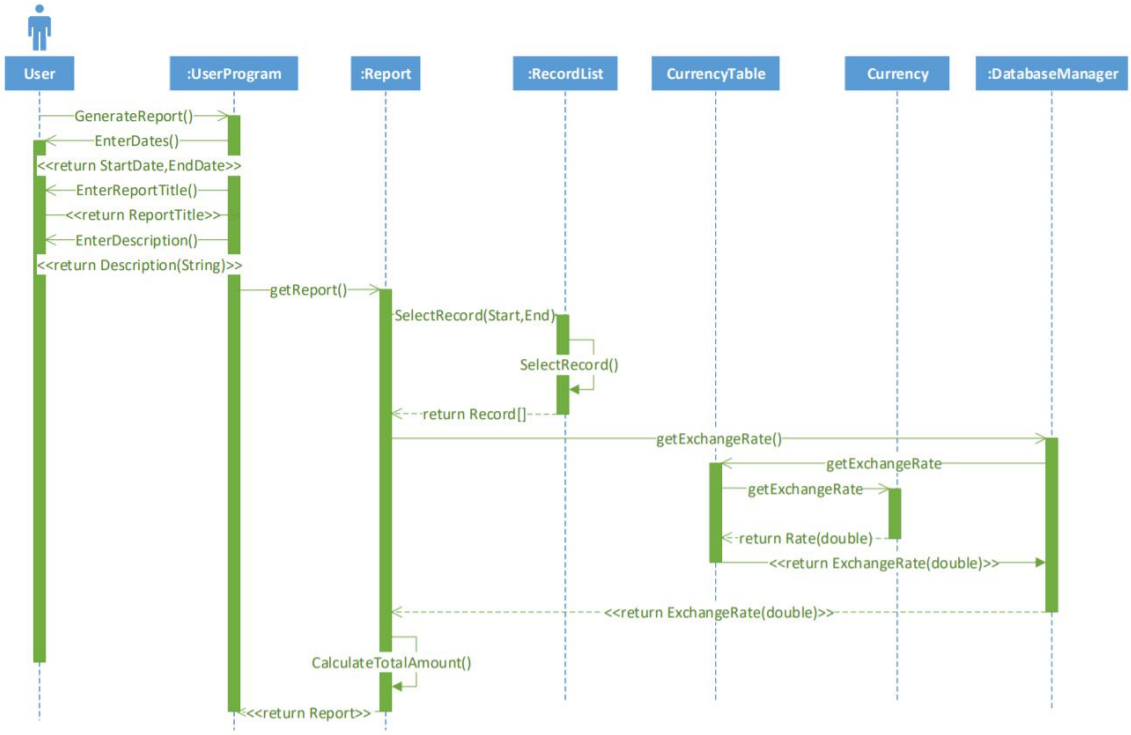
CreateRecord



This sequence diagram shows the process of create record, users should provide some basic information to this expense this time (on the sequence diagram like consumption date, category and cost of each items) to user program. User program collect these information to create record and then add them into the record list. After recording, return the information to user program.

User can abort record by sending the stop information to user program.

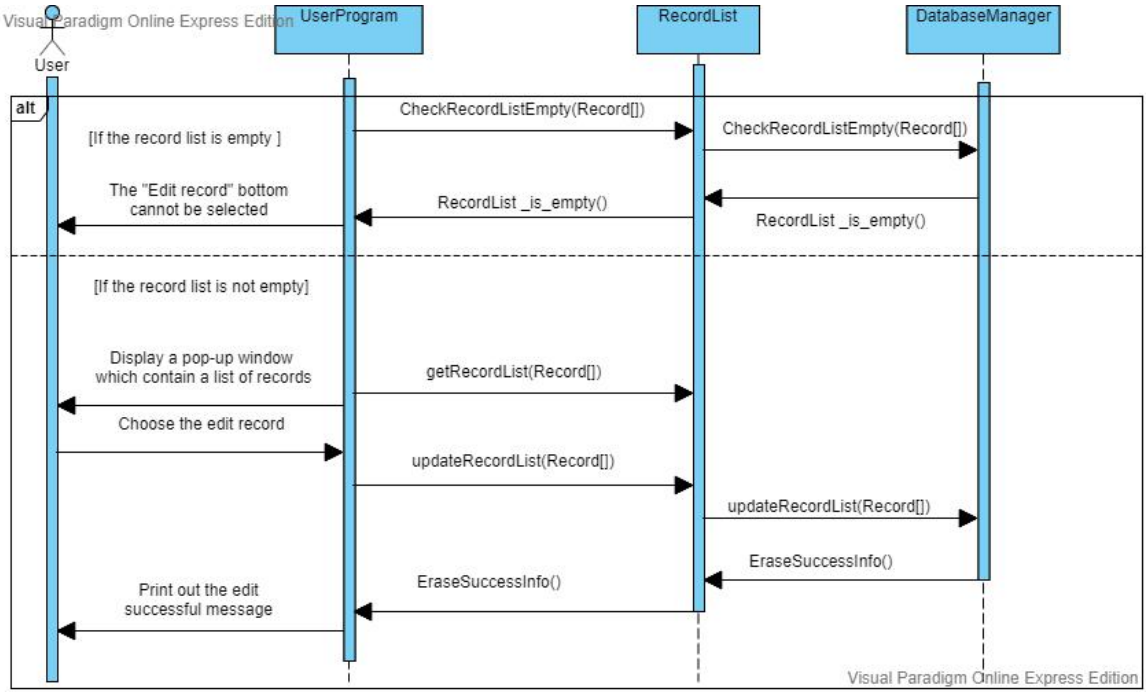
ExpenseSummary



This sequence diagram is about record generation. It allowed user to get the record list and summary in a period of time. User should provide the time slot to user program(it also allowed user to give a title and description to this record list). User program send message to “Report” and “Report” gain message from “RecordList”.

After getting all of the data, “Record” will visit Database Manager if their exist a money exchange. Database Manager gain the currency exchange rate from Currency table and return to ”Report”, “Report” calculate the total amount and return to User Program.

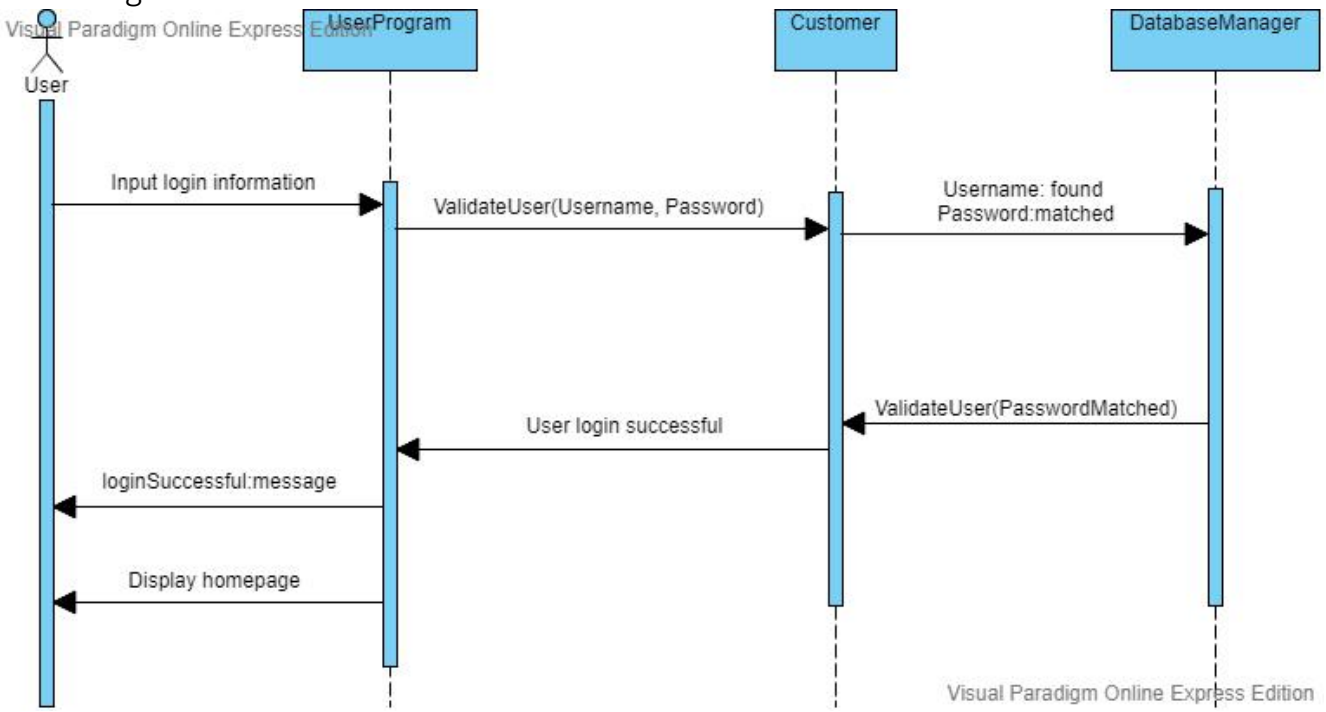
EditRecord



The Sequence diagram “Edit Record” shows the way to erase the appoint record.

User Program check if the list is empty if user want to edit some records from Record List, and the “RecordList” visit database to make sure the record list is empty or not. If the record list is empty, user program return “The Edit record bottom cannot be selected” and sequence end. when the record list is not empty, “UserProgram” go back to “RecordList” ,gain all of records and display it to user. In this time, users need to select the record which they want to edit in “UserProgram”. “UserProgram” give a feedback to “RecordList” and “RecordList” upload new data to database. When uploading successful, user can get a success tips from “UserProgram”.

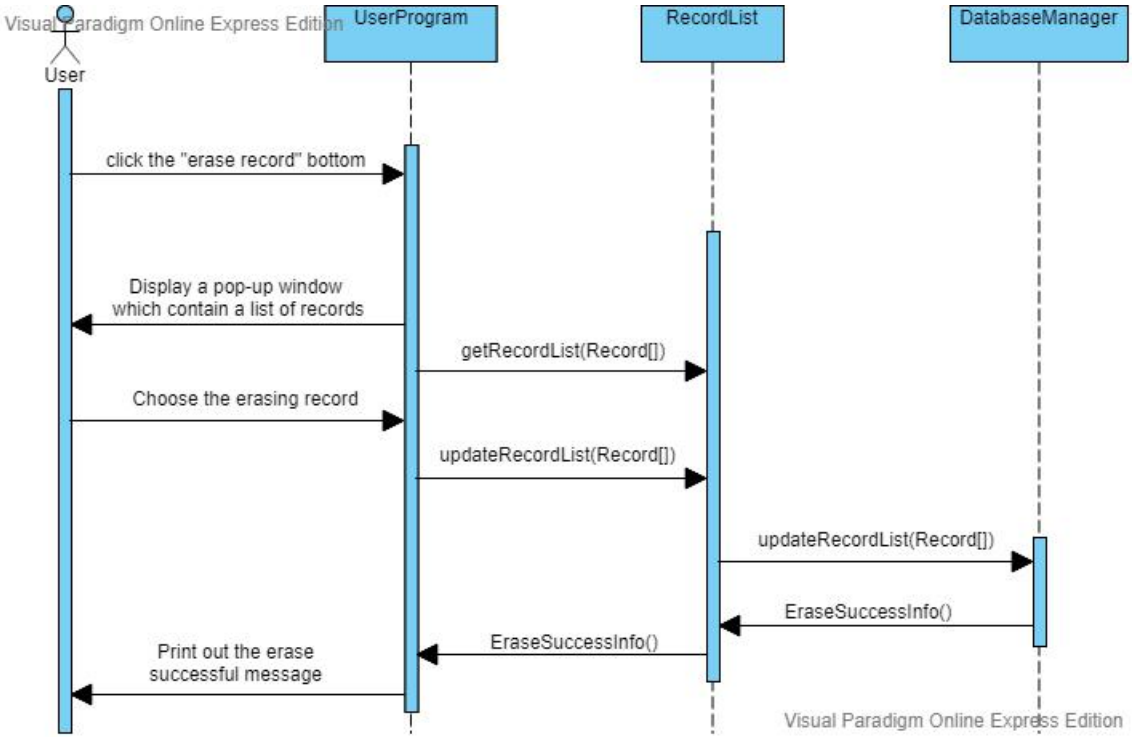
Login



This Sequence diagram shows the function Login

User enter the username and password to “UserProgram” , “UserProgram” send the data to “Customer” to check if the username and password is valid. Customer will bring them to Database to check can username be found and whether the password can be matched If it is a valid username-password.(return fail if the information provided is invalid or username can not be found). When all things correct, “UserProgram” gonna have a login Successful feedback from Customer(from Database Manager) and “UserProgram” then display homepage

EraseRecord



The Sequence diagram “EraseRecord” shows the way to erase the appoint record.

User Program check if the list is empty if user want to erase some records from Record List, and the “RecordList” visit database to make sure the record list is empty or not. If the record list is empty, user program return “The Edit record bottom cannot be selected” and sequence end. when the record list is not empty, “UserProgram” go back to “RecordList” ,gain all of records and display it to user. In this time, users need to select the record which they want to erase in “UserProgram”. “UserProgram” give a feedback to “RecordList” and “RecordList” upload new data to database. When uploading successful, user can get a success tips from “UserProgram”.