MICROSOLUTIONS

TECHNOLOGY SOLUTIONS

Phase 2

2019-11-24

Version 2

Prepared by group members:

Al-Safadi Hadeal	104 601 675
Dube Lincoln	104 562 749
Elchami Mohamad	104 902 736
Fuller David	104 657 834
Tanimowo Olushola	104 606 311
Vo Vivian	104 769 052
Wu James	104 038 826

Table of Contents

Revision History

Version	Date	Name	Description
1	11/07/19	Phase 2	
2	11/24/19	Phase 2	

1 Phase 2 Task 1:

1.1 Functional Requirements

ID	**Description**	**Priority**
RQ1	**The system should allow users to create an Account**	**High**
RQ2	**The system should allow users to transfer money**	**High**
RQ3	**The system should allow users to pay bills**	**High**
RQ4	**The system should allow the user to update their private information.**	**Medium**
RQ5	**The system should allow user to add expenses.**	**High**
RQ6	**The system should allow user to add income.**	**High**
RQ7	**The system should allow user to search data they need.**	**Medium**
RQ8	**The system should allow user to remove expenses.**	**Medium**
RQ9	**The system should allow user to remove income.**	**Medium**

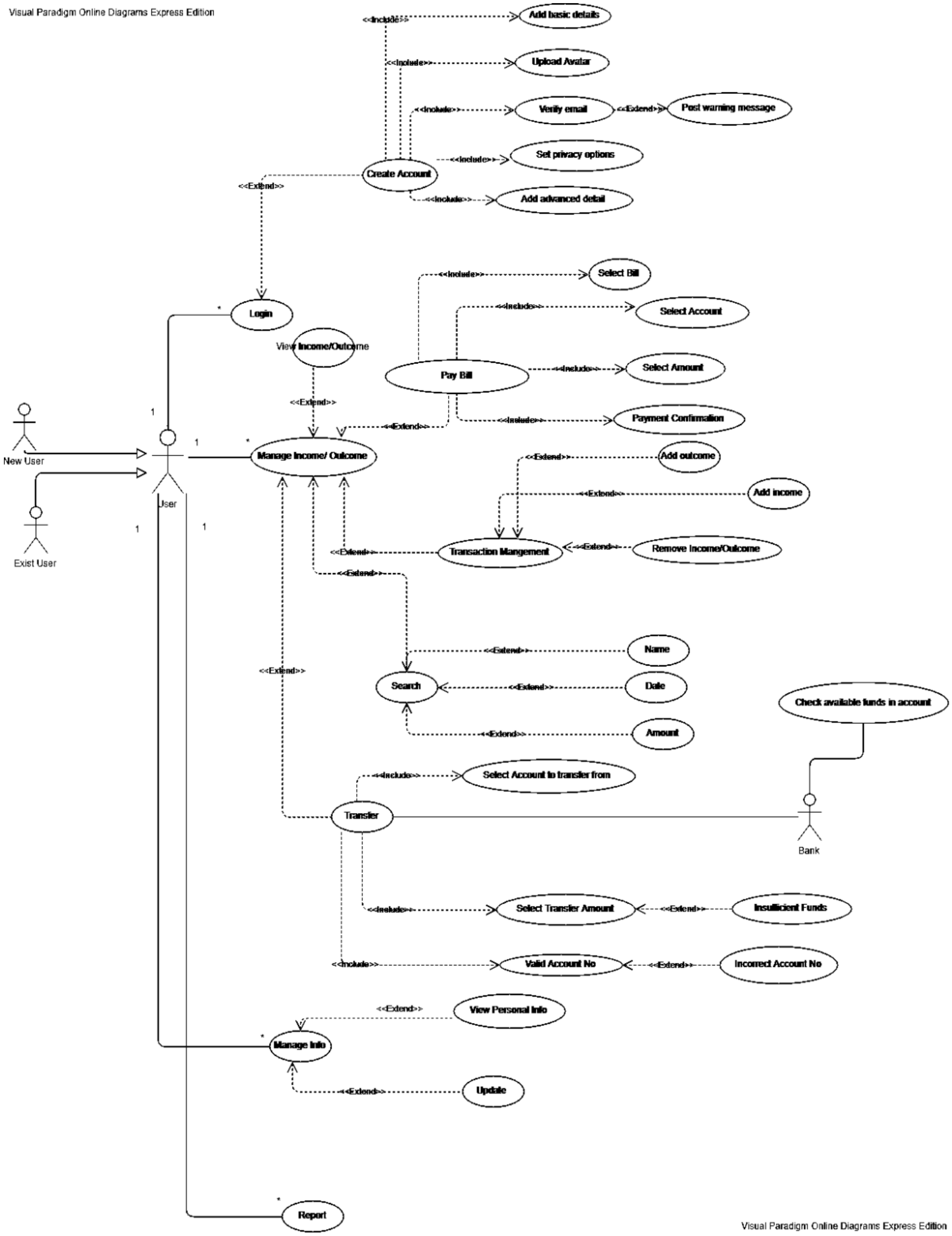
RQ10	The system should allow user to view the monthly and annually report.	Medium

1.2 Non-functional Requirements

ID	Description	Priority
RQ11	The system should run on tablet PCs, mobile, or tablet.	Low
RQ12	User must be able to view current expenditure	High
RQ13	Only the user and system can see users records and data.	High
RQ14	The system should be able to work on any OS environment	High
RQ15	The interaction between user and the system should not exceed 5 seconds	High
RQ16	The system and user both must have a connection to the internet	High

2 Phase 2 Task 2:

2.1 Use Case Diagram



2.2 Use Case Description

2.2.1 RQ1: Create Account

<table>
<tr><td>Use Case Name: Create account</td><td>ID: RQ1</td><td>Priority: High</td></tr>
<tr><td colspan="3">Actors: User</td></tr>
<tr><td colspan="3">Description: The Create Account use case allows the User to create a login and become a Registered User.</td></tr>
<tr><td colspan="3">Trigger:
Type: ☑External □Temporal</td></tr>
<tr><td colspan="3">Preconditions:</td></tr>
<tr><td colspan="2">Normal Course:
This use case starts when the User accesses the system feature that enables him/her to create an account by entering information that is maintained in the User’s account.
The User enters the required User Account information values and requests that the system saves the entered values.
The system validates the entered User Account information.
The values for the User Account information are stored in the User’s account. The system notifies the User that the account has been created.
The use case ends</td><td>Information of Steps:</td></tr>
</table>

Alternative Courses: At any time, the User may choose to cancel the account creation. At which point, the processing is discontinued, the user account remains unchanged, and the user is notified that the account management request has been cancelled.	
If during Create Account, the system determines that the User entered invalid User Account information, the following occurs: The system describes which entered data was invalid and presents the User with suggestions for entering valid data. The system prompts the User to re-enter the invalid information. The User re-enters the information and the system re-validate it. If valid information is entered, the User Account Information is stored. If invalid information is entered, the Entered Information is Invalid alternative flow is executed again. This continues until the User enters valid information or chooses Cancel (see the User Cancels Account Management Request alternative flow).	Invalid User Account information: Missing information items Username already exists in the system User Account information entered does not comply to its definition in the glossary Not well formed e-mail address Offending words in any part of the User Account information
PostConditions: If successful, the User entered data is stored in the user account. Confirmation is sent to the appropriate email address. If unsuccessful, the User entered invalid data or chose to cancel the account creation request. In either case, no account will be created.	
Exceptions: None	

2.2.2 RQ2: Transfer Funds

Use Case Name: Transfer Funds	ID: RQ2	Priority: High
Actors: User, Bank		
Description:The User should be able to use the application to transfer between owned bank accounts		
Trigger: User wants to transfer funds between accounts Type: ☑External □Temporal		
Preconditions: User is authenticated and connected to the internet		
Normal Course: 1. User requests a transfer 2. User chooses source bank account 3. User chooses destination bank account 4. User chooses amount 5. User confirms transfer 6. System transfers funds 7. System records transfer		Information of Steps: 1. NA 2. BankAccountID 3. BankAccountID 4. TransferAmount 5. NA 6. 2, 3, 4 7. 2, 3, 4
Alternative Courses: 4.1. Insufficient Funds 4.2. Cancel Transfer 6.1 Attempt to access BankAccount fails due to invalid Account		4.1. BankAccountID 4.2. NA 6.1 BankAccountID
PostConditions: Funds are transferred between accounts and a record has been made.		
Exceptions: E1: BankAccount is invalid. E2: No funds available		

2.2.3 RQ3: Bill Payment

<table>
<tr><td>Use Case Name: Bill payment</td><td>ID: RQ3</td><td>Priority: Moderate</td></tr>
<tr><td colspan="3">Actors: User, system</td></tr>
<tr><td colspan="3">Description: This use case describes how the user pay bills</td></tr>
<tr><td colspan="3">Trigger: A pending bill payment is created, and the user is notified
Type: ☑ External □Temporal</td></tr>
<tr><td colspan="3">Preconditions: A pending bill is created in the datastore</td></tr>
<tr><td colspan="2">Normal Course:
1. The user retrieves the pending bill payment from the database
2. The system prompts the user to enter the amount to pay
3. The system prompts the user to confirm or cancel payment</td><td>Information of Steps:
1. Pending bill id
2. Account id and amount
3. Offer decision</td></tr>
<tr><td colspan="2">Alternative Courses:
1. The user doesn’t have enough money and attempts to pay the bill.
2. System display message there is no enough money
3. The system prompts the user to add funds</td><td>1. Pending bill id
2. Account id</td></tr>
<tr><td colspan="3">PostConditions:
1. Bill payment is recorded in the pending payment datastore
2. Pending bill payment is removed from pending payment datastore and added to accept payment or rejected payment</td></tr>
<tr><td colspan="3">Exceptions:
E1. User login unsuccessful
1. System display unsuccessful login
2. Require user to re-login</td></tr>
</table>

2.2.4 RQ4: Manage Information

Use Case Name: Manage Information	ID: RQ4	Priority: High
Actors: User, System		
Description: This Use Case interacts with the System to apply changes (update) to the user information.		
Trigger: User wants to use the app to pay bill. Type: ☑External □Temporal		
Preconditions: 1. The agent must log into the system first		
Normal Course: 1. The Use Case begins when a request to update an account information is received. 2. The system will ask what information users want to update and require user to enter the new data. 3. User enters input. 4. The Accounting System will send a status of OK. 5. The Use Case ends.		Information of Steps:
Alternative Courses: None		
PostConditions: If successful: New information is saved and stored in the database. If it fails: The system notify user the errors.		
Exceptions: E. Null information System requires user to re-enter data want to update.		

2.2.5 RQ5: Add Expense

Use Case Name: Add Expense	ID: RQ5	Priority: Medium
Actors: User		
Description: In this Use Case user interacts with the System to add new expense.		
Trigger: User wants to use the app to add expenses. Type: ☑External □Temporal		
Preconditions: The user must log into the system first and go to money management function successfully.		
Normal Course: 1. After user has logged into their account. They redirect to Money Management. The Use Case begins when a request to add expense to account. 2. The system require user to enter details the data. 3. User enters input. 4. The System will send a status of OK if the input is valid, else return error notification. 5. The Use Case ends.		Information of Steps:
Exceptions: E. Null information System requires user to re-enter the amount of expense user want to add.		

2.2.6 RQ6: Add Income

Use Case Name:Add Income	ID: RQ6	Priority: Medium
Actors: User		
Description: In this Use Case user interacts with the System to add new income.		
Trigger: User wants to use the app to add income. Type: ☑External □Temporal		
Preconditions: The user must log into the system first and go to money management function successfully.		
Normal Course: 1. After user has logged into their account. They redirect to Money Management. The Use Case begins when a request to add income to account. 2. The system require user to enter details the data. 3. User enters input. 4. The System will send a status of OK if the input is valid, else return error notification. 5. The Use Case ends.		Information of Steps:
Exceptions: E. Null information System requires user to re-enter the amount of income user want to add.		

2.2.7 RQ7: Search

Use Case Name: Search	ID: RQ7	Priority: High
Actors: User, System		
Description: This Use Case describes the process for finding a particular data in the system.		
Trigger: User search data they want to know (by amount of money, by date, by name). Type: ☑External □Temporal		
Preconditions: System must start successful, no errors.		
Normal Course: 1. The Use Case starts when the Search is displayed. 2. The user selects Search by date, amount, or name. 3. The user selects search data that they want. 4. If the user entered a name 4.1 The system will display a list of data for that customer, including at least an amount and date of order 4.2 The user will select one end if 5. The system returns an order id and the Use Case ends.		Information of Steps:
Alternative Courses: The system will give an error if the user has not selected a time period. The system will give an error if the user enters the wrong time period.		
PostConditions: If the use case is successfully implemented, the statistics table of transaction information by time period will be displayed, otherwise the system status is unchanged.		
Exceptions: E. Not enough money in the accounts 1. System display message there is not enough money 2. System ask user to add money		

2.2.8 RQ8: Remove Expense

Use Case Name: Remove Expense	ID: RQ8	Priority: Medium
Actors: User, System		
Description: The system to allow users to remove expenses		
Trigger: User wants to remove expense from data storage Type: ☑External ☐Temporal		
Preconditions: User is authenticated and connected to the internet		
Normal Course: 1. User clicks remove expense 2. User chooses expense to remove 3. System prompts the user for confirmation 4. User chooses yes 5. System prompt the user if they want to remove more expense i. If the user chooses yes, user goes back to number 2. ii. If user chooses no, user is taken to main menu		Information of Steps: 1. NA 2. ExpenseID 3. NA 4. NA 5. NA
Alternative Courses: 6. No expense to remove 7. System prompts user of lack of expense 8. User taken back to main menu		6. NA 7. NA 8. NA
PostConditions: 1. Expense is removed from the data storage. 2. System prompts a message of expense remover.		

Exceptions:
E1: BankAccount is invalid.

2.2.9 RQ9: Remove Income

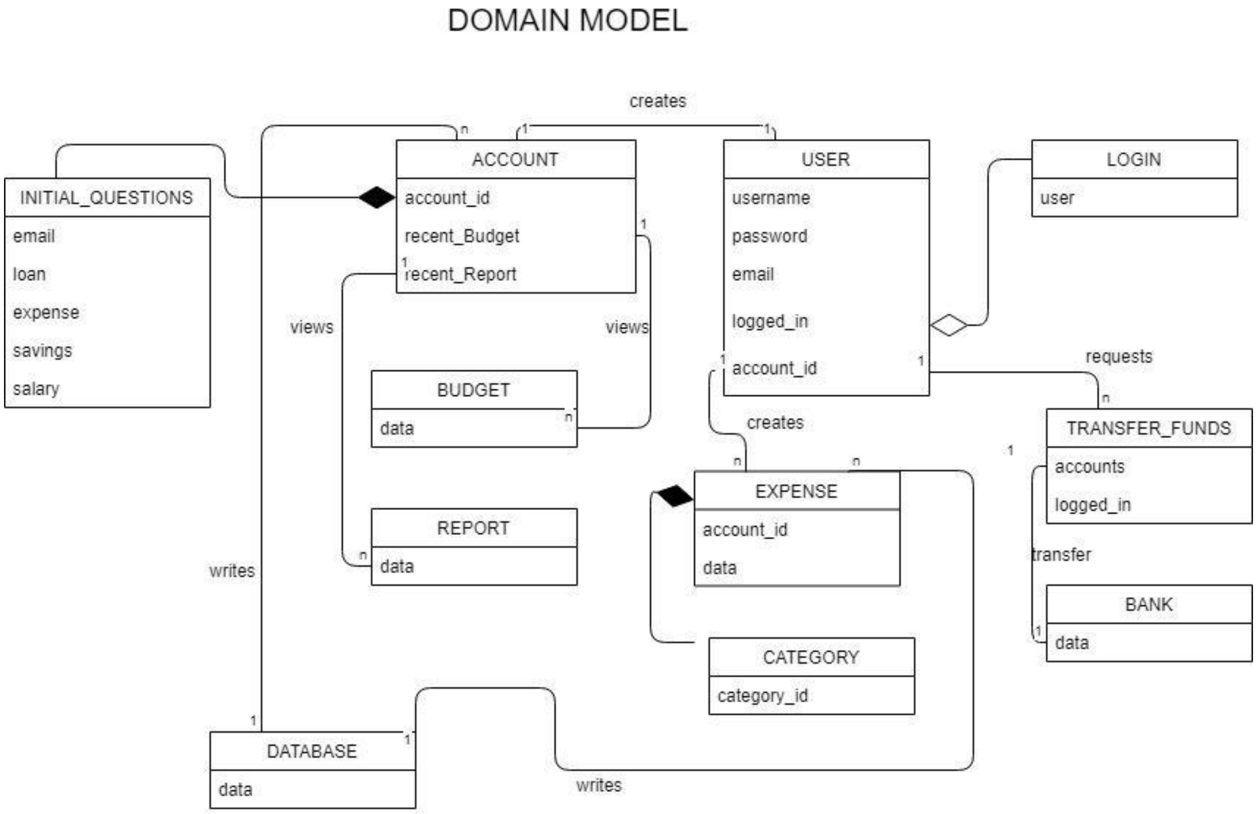
Use Case Name: Remove Income	ID: UC9	Priority: Medium
Actors: User		
Description: In this Use Case user interacts with the System to add remove income.		
Trigger: User wants to use the app to remove income. Type: ☑External □Temporal		
Preconditions: The user must log into the system first and go to money management function successfully. Having at least one income stored in their system.		
Normal Course: 1. After user has logged into their account. They redirect to Money Management. The Use Case begins when a request to remove income of the account. 2. The system require user to enter details the data. 3. User enters input. 4. The System will send a status of OK if the input is valid, else return error notification. 5. The Use Case ends.		Information of Steps:
Exceptions: E. Null information System requires user to re-enter the amount of income user want to remove.		

2.2.10 RQ10: Report

Use Case Name: Report	ID: RQ10	Priority: High
Actors: User		
Description: The actor uses this Use Case to get monthly or annually report.		
Trigger: User wants to use the financial report. Type: ☑External □Temporal		
Preconditions: 1. The user must log into the system first		
Normal Course: 1. The actor chooses the monthly or annually report. 2. The system returns the data in report form 3. Use case ended		Information of Steps:
Alternative Courses:1. Users cancel getting the report. 2. The system skips the document and returns to the main interface. 3.Usecase ended		
PostConditions: If successful: Display the list of expenses and income data If it fails: Notify the user about the error		
Exceptions: None		

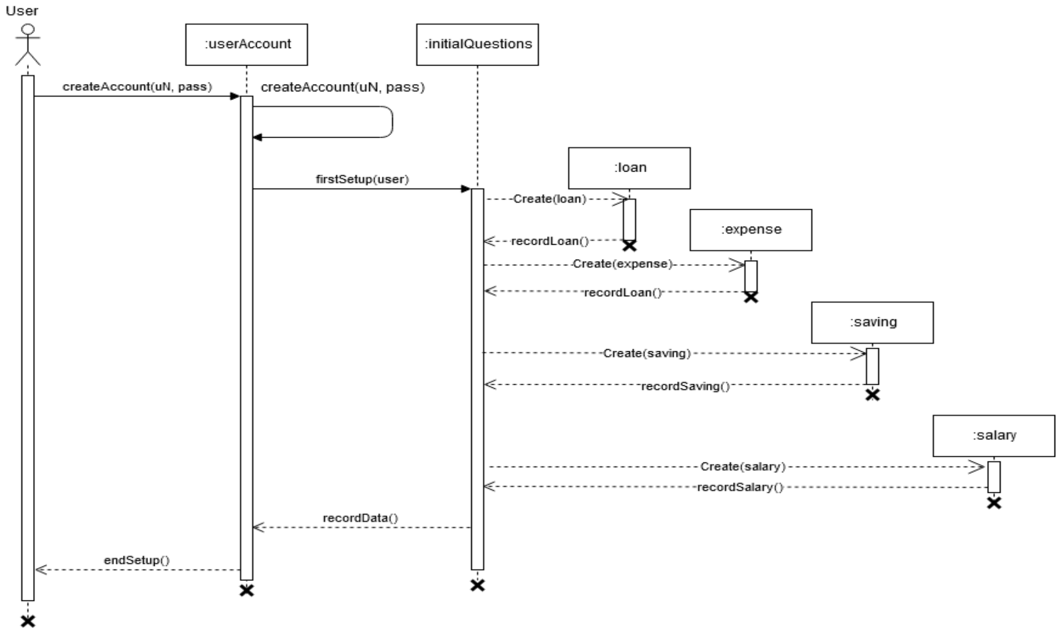
3 Phase 2 Task 3:

3.1 Domain Model

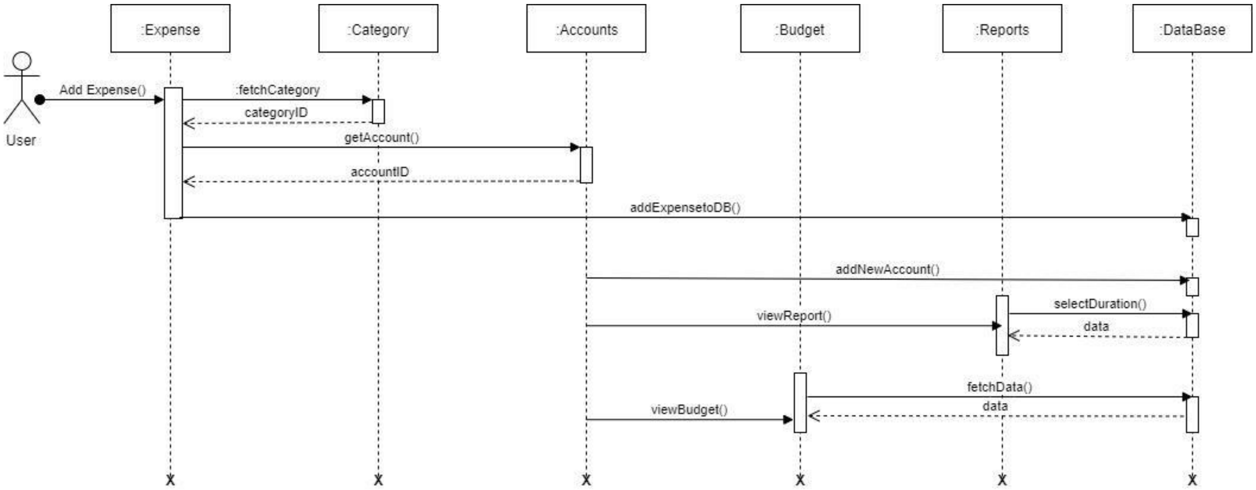


4 Phase 2 Task 4:

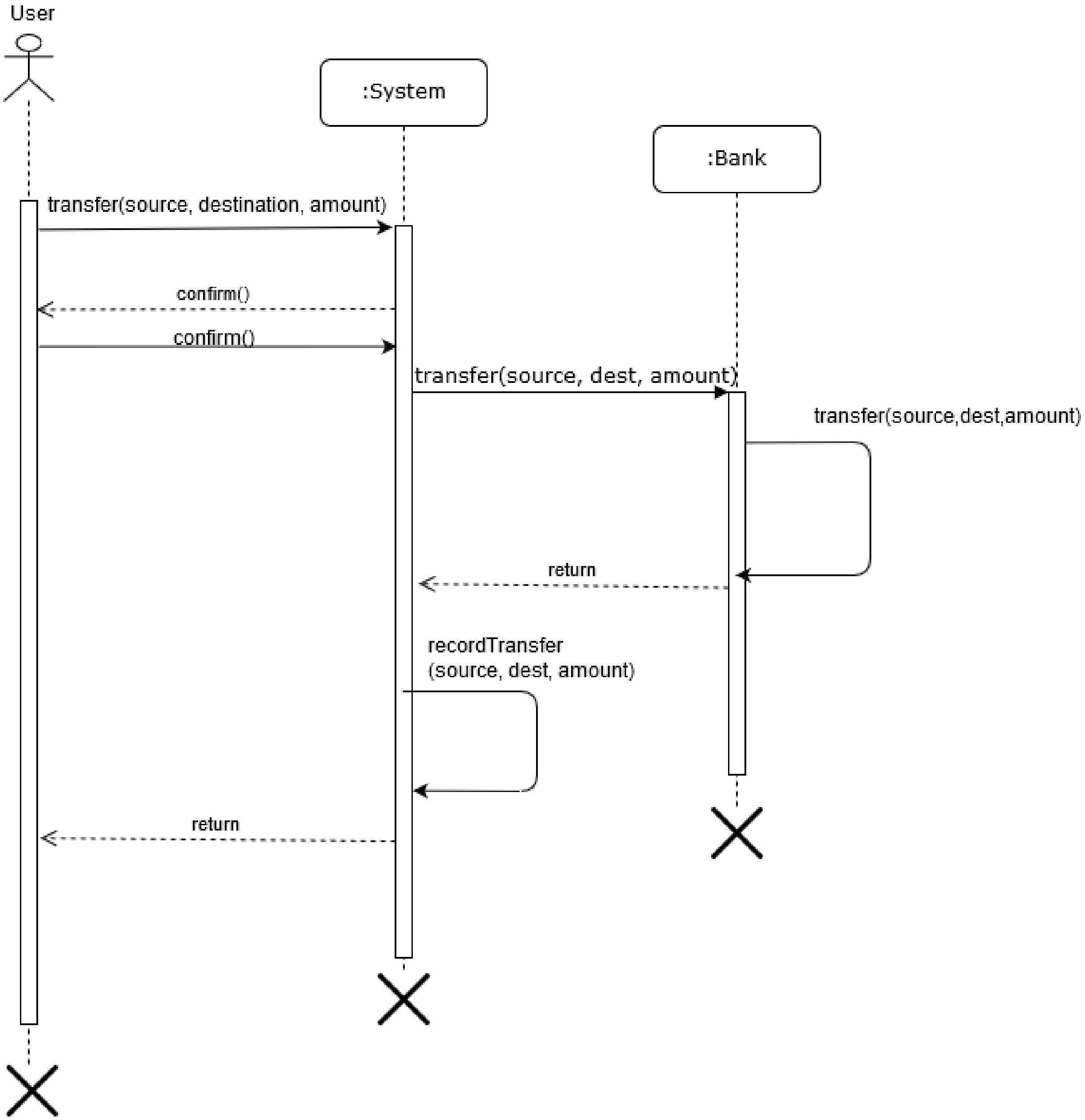
4.1 First Time Setup - Register and Initial Questions:



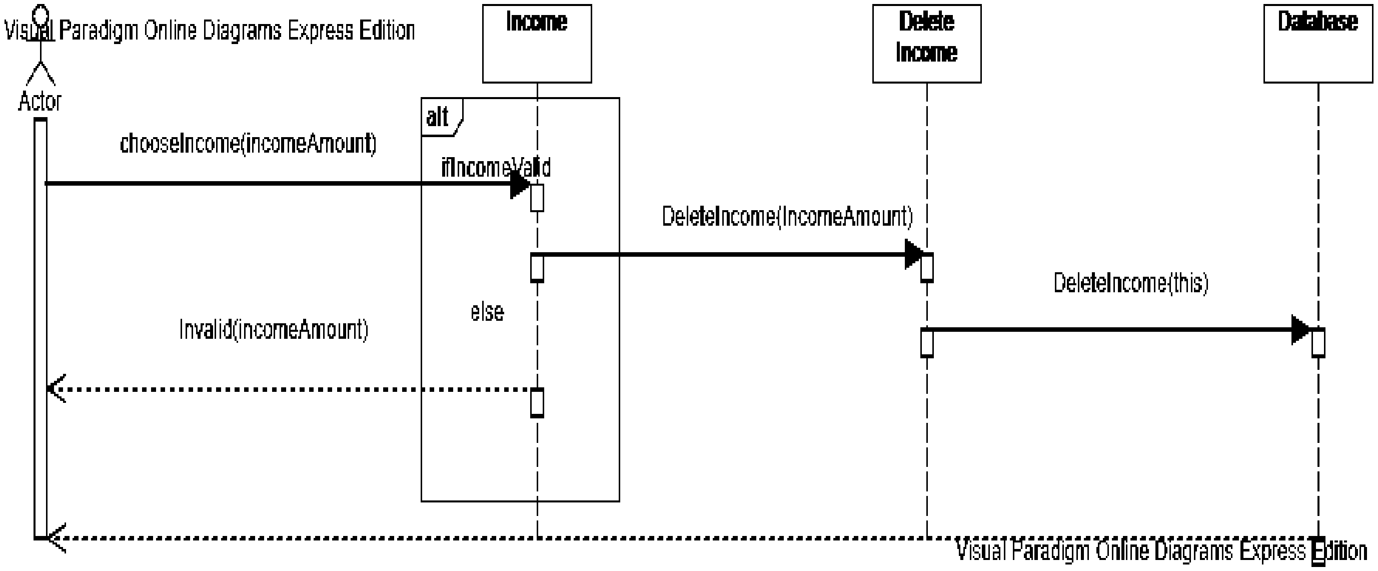
4.2 Manage Expenses



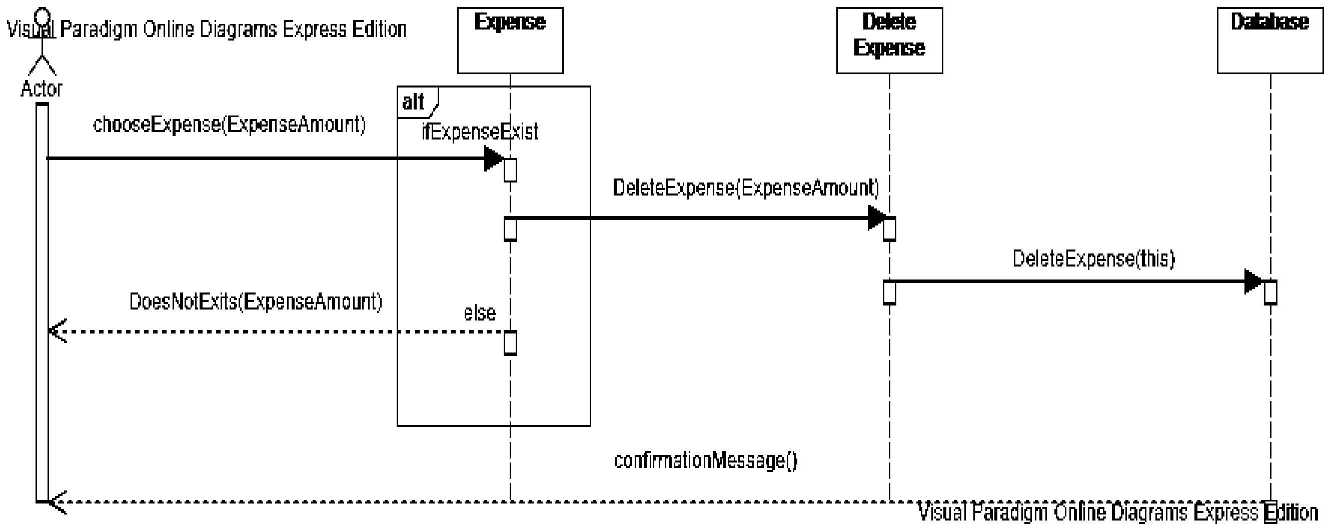
4.3 Transfer Funds


User
:System
:Bank
transfer(source, destination, amount)
confirm()
confirm()
transfer(source, dest, amount)
transfer(source,dest,amount)
return
recordTransfer
(source, dest, amount)
return

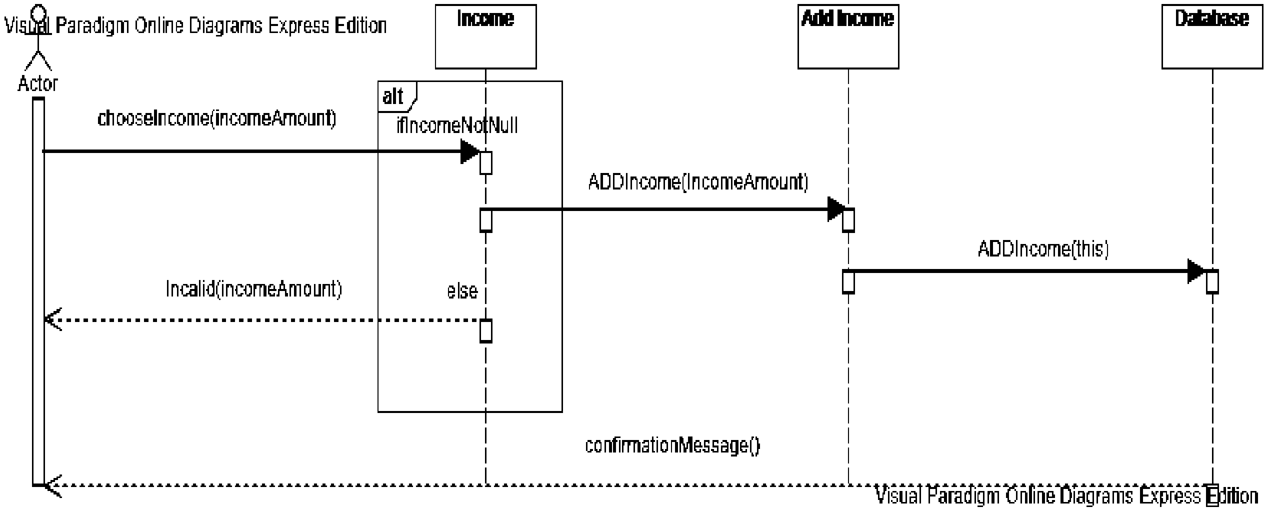
4.4 Delete Income


Visual Paradigm Online Diagrams Express Edition
Actor
Income
Delete Income
Database
alt
chooseIncome(incomeAmount)
ifIncomeValid
DeleteIncome(IncomeAmount)
DeleteIncome(this)
else
Invalid(incomeAmount)
Visual Paradigm Online Diagrams Express Edition

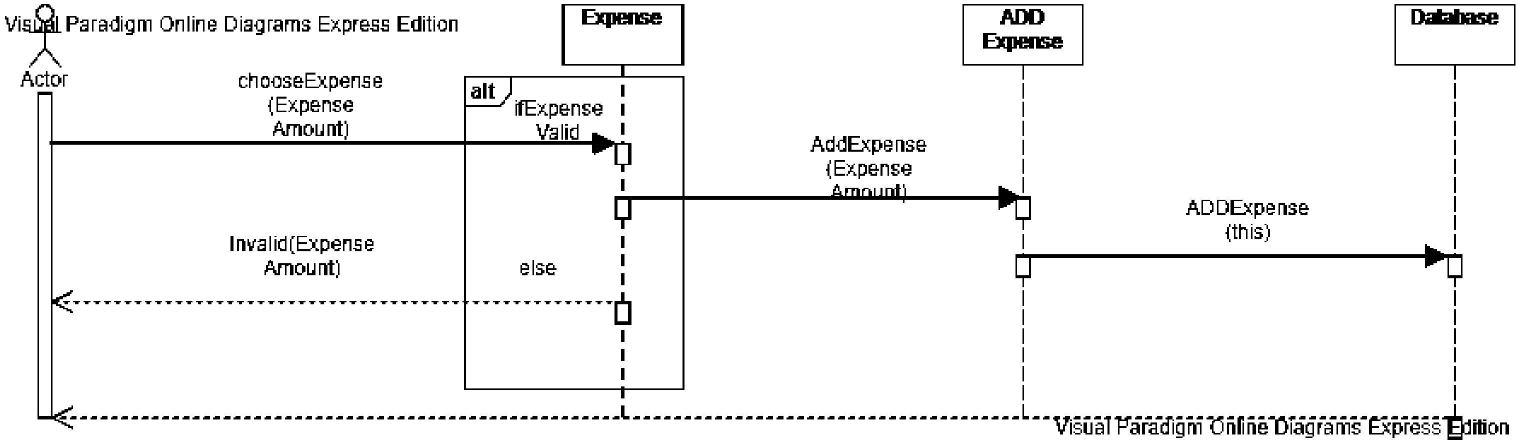
4.5 Delete Expense


Visual Paradigm Online Diagrams Express Edition
Actor
Expense
Delete Expense
Database
alt
chooseExpense(ExpenseAmount)
ifExpenseExist
DeleteExpense(ExpenseAmount)
DeleteExpense(this)
DoesNotExits(ExpenseAmount)
else
confirmationMessage()
Visual Paradigm Online Diagrams Express Edition

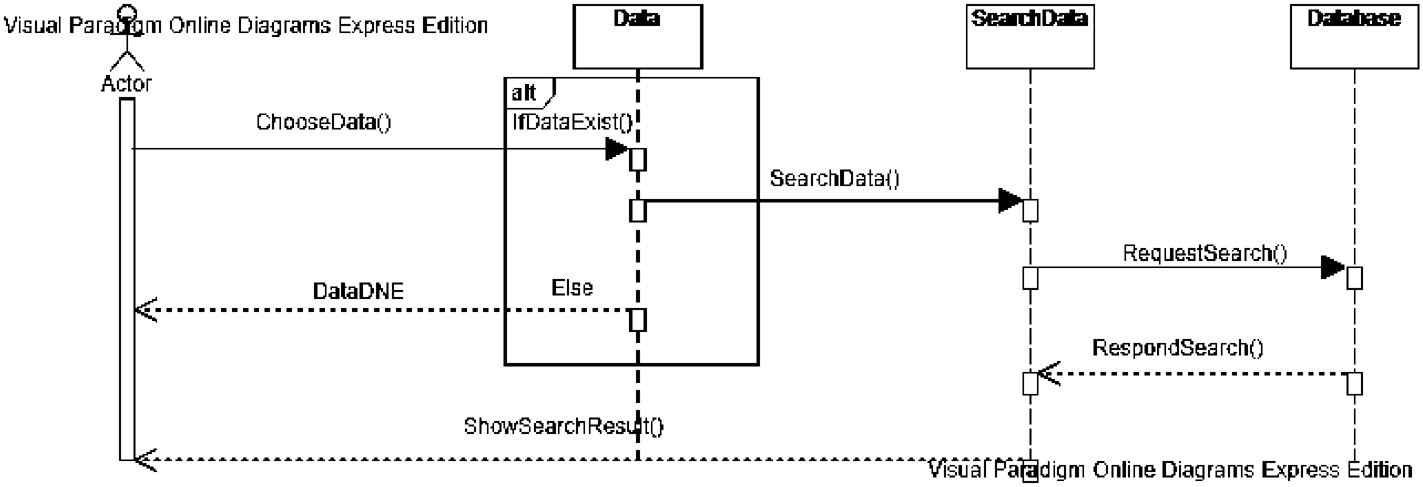
4.6 Add Income



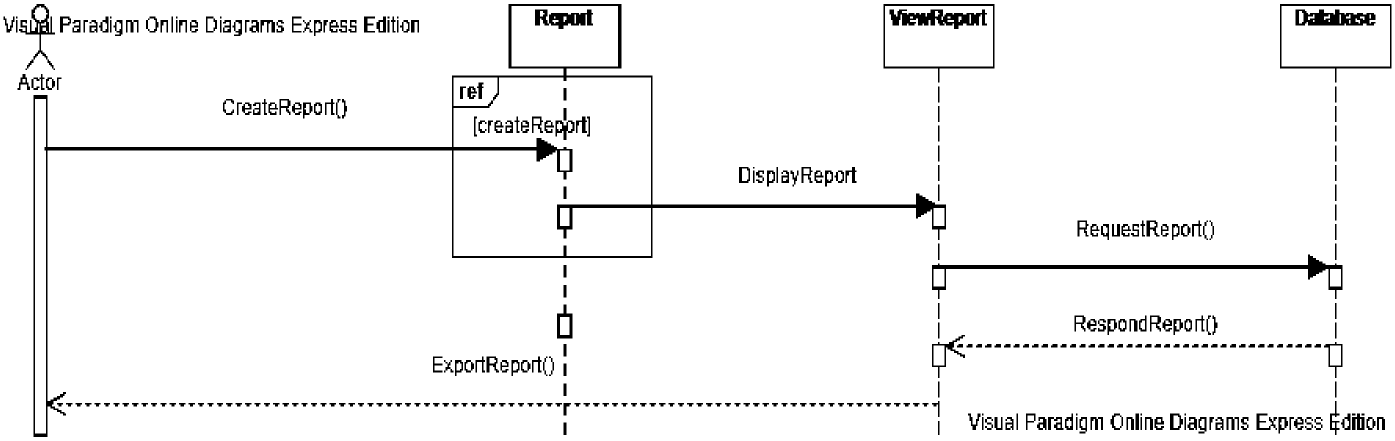
4.7 Add Expense


Visual Paradigm Online Diagrams Express Edition
Actor
Expense
ADD Expense
Database
alt
chooseExpense (Expense Amount)
ifExpense Valid
AddExpense (Expense Amount)
ADDExpense (this)
Invalid(Expense Amount)
else
Visual Paradigm Online Diagrams Express Edition

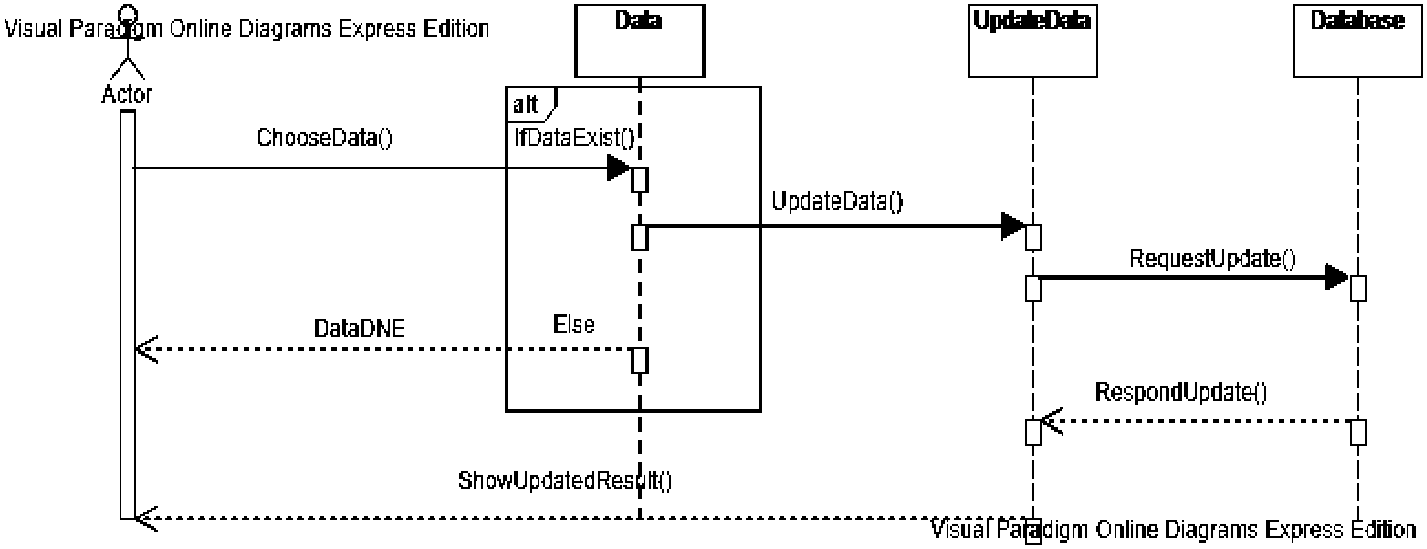
4.8 Search


Visual Paradigm Online Diagrams Express Edition
Actor
Data
SearchData
Database
alt
ChooseData()
IfDataExist()
SearchData()
RequestSearch()
DataDNE
Else
RespondSearch()
ShowSearchResult()
Visual Paradigm Online Diagrams Express Edition

4.9 Report


Visual Paradigm Online Diagrams Express Edition
Actor
Report
ViewReport
Database
ref
[createReport]
CreateReport()
DisplayReport
RequestReport()
RespondReport()
ExportReport()
Visual Paradigm Online Diagrams Express Edition

4.10 Update Information


Visual Paradigm Online Diagrams Express Edition
Actor
Data
UpdateData
Database
alt
ChooseData()
IfDataExist()
UpdateData()
RequestUpdate()
DataDNE
Else
RespondUpdate()
ShowUpdatedResult()
Visual Paradigm Online Diagrams Express Edition