**Infy Credit Card Inc.**

**(ICCI)**

**Detailed Design**

****

**Education & Research Department**

**© Infosys Technologies Ltd**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version Revision** | **Date** | **Author** | **Reviewed by** | **Description** |
| 1.00 | 22 Apr, 2010 | Ashwini K J, Deepak M, Ramya M S, Vani Gopalkrishnan |  | Initial write |
| 1.00a | 25 Apr, 2010 | Ashwini K J |  | Incorporated the review comments |
| 2.00 | 4 Jun. 10 | Ashwini K J, Deepak M, Ramya M S, Vani Gopalkrishnan |  | Added modules |

*Table Of Contents*

1. Overview 5

2. High level design architecture 6

2.1 Subsystems 6

2.2 Modeling 7

2.3 Layering and Partitioning 7

2.4 Categorization of classes 7

2.5 Coding and UI standards 7

2.6 System test planning 7

2.7 Development environment 7

2.8 Software 8

2.9 Hardware/software and platform 8

2.10 Components off the Shelf (COTS) 8

3. Detailed Design 8

3.1 Implementation elements 8

3.2 Design of algorithms 9

3.3 Database design 9

3.4 Unit test planning 9

3.5 Optimization of algorithms and data access 9

3.6 External interfaces design 9

3.7 Error messages 9

3.8 Screen layout 9

3.9 Source for output fields 9

3.10 Special processing notes 10

3.10.1 Assumptions 10

3.10.2 Limits 10

3.10.3 Exception handling 10

4. class design specification 10

4.1 ENTITY BEANS 10

4.1.1 BlockedCard Entity Bean Class 10

4.1.1.1 Class Identification 10

4.1.1.2 Attributes (or Data members) 11

4.1.1.3 Methods (or class functions) 11

4.1.2 Card Application Entity Bean Class 11

4.1.2.1 Class Identification 11

4.1.2.2 Attributes (or Data members) 11

4.1.2.3 Methods (or class functions) 11

4.1.3 Card Entity Bean Class 12

4.1.3.1 Class Identification 12

4.1.3.2 Attributes (or Data members) 12

4.1.3.3 Methods (or class functions) 12

4.1.4 Customer Entity Bean Class 12

4.1.4.1 Class Identification 12

4.1.4.2 Attributes (or Data members) 12

4.1.4.3 Methods (or class functions) 13

4.1.5 Payment Entity Bean Class 13

4.1.5.1 Class Identification 13

4.1.5.2 Attributes (or Data members) 13

4.1.5.3 Methods (or class functions) 13

4.1.6 Scheme Entity Bean Class 13

4.1.6.1 Class Identification 13

4.1.6.2 Attributes (or Data members) 14

4.1.6.3 Methods (or class functions) 14

4.1.7 Transaction Entity Bean Class 14

4.1.7.1 Class Identification 14

4.1.7.2 Attributes (or Data members) 14

4.1.7.3 Methods (or class functions) 15

4.1.8 Login Entity Bean Class 15

4.1.8.1 Class Identification 15

4.1.8.2 Attributes (or Data members) 15

4.1.8.3 Methods (or class functions) 15

4.2 TRANSFER OBJECTS 15

4.2.1 BlockedCardTO Class 15

4.2.1.1 Class Identification 15

4.2.1.2 Attributes (or Data members) 16

4.2.1.3 Methods (or class functions) 16

4.2.2 CardTO Class 16

4.2.2.1 Class Identification 16

4.2.2.2 Attributes (or Data members) 16

4.2.2.3 Methods (or class functions) 16

4.2.3 CardApplicationTO 17

4.2.3.1 Class Identification 17

4.2.3.2 Attributes (or Data members) 17

4.2.3.3 Methods (or class functions) 17

4.2.4 CustomerTO Class 17

4.2.4.1 Class Identification 17

4.2.4.2 Attributes (or Data members) 17

4.2.4.3 Methods (or class functions) 18

4.2.5 Payment TO Class 18

4.2.5.1 Class Identification 18

4.2.5.2 Attributes (or Data members) 18

4.2.5.3 Methods (or class functions) 18

4.2.6 SchemeTO Class 18

4.2.6.1 Class Identification 18

4.2.6.2 Attributes (or Data members) 19

4.2.6.3 Methods (or class functions) 19

4.2.1 TransactionTO 19

4.2.1.1 Class Identification 19

4.2.1.2 Attributes (or Data members) 19

4.2.1.3 Methods (or class functions) 20

4.2.8 Login Transfer Object Class 20

4.2.8.1 Class Identification 20

4.2.8.2 Attributes (or Data members) 20

4.2.8.3 Methods (or class functions) 20

4.2.1 ApprovedCardTO Class 20

4.2.1.1 Class Identification 20

4.2.1.2 Attributes (or Data members) 21

4.2.1.3 Methods (or class functions) 21

4.3 DATA SERVICES 21

4.3.1 BlockedCardService Class 21

4.3.1.1 Class Identification 21

4.3.1.2 Attributes (or Data members) 21

4.3.1.3 Methods (or class functions) 21

4.3.1.4 Attributes (or Data members) 21

4.3.1.5 Methods (or class functions) 22

4.3.1.6 Attributes (or Data members) 22

4.3.1.7 Methods (or class functions) 22

4.3.1.8 Attributes (or Data members) 22

4.3.2 CardService Class 22

4.3.2.1 Class Identification 22

4.3.2.2 Attributes (or Data members) 23

4.3.2.3 Methods (or class functions) 23

4.3.2.4 Attributes (or Data members) 23

4.3.2.5 Attributes (or Data members) 23

4.3.3 CustomerService Class 24

4.3.3.1 Class Identification 24

4.3.3.2 Attributes (or Data members) 24

4.3.3.3 Methods (or class functions) 24

Methods (or class functions) 24

4.3.4 SchemeService Class 25

4.3.4.1 Class Identification 25

4.3.4.2 Attributes (or Data members) 25

4.3.4.3 Methods (or class functions) 25

4.3.4.4 Attributes (or Data members) 25

4.3.3.5Methods (or class functions) 25

4.3.4.6 Attributes (or Data members) 25

4.3.3.7 Methods (or class functions) 26

4.3.3.8Attributes (or Data members) 26

Methods (or class functions) 26

Methods (or class functions) 26

4.3.5 PaymentService Class 27

4.3.5.1 Class Identification 27

4.3.5.2 Attributes (or Data members) 27

4.3.5.3 Methods (or class functions) 27

4.3.5.4 Methods (or class functions) 27

4.3.6 Transaction Service Class 27

4.3.6.1 Class Identification 27

4.3.6.2 Attributes (or Data members) 28

4.3.6.3 Methods (or class functions) 28

4.3.7 Login Service Class 29

4.3.7.1 Class Identification 29

4.3.7.2 Attributes (or Data members) 29

4.3.7.3 Methods (or class functions) 29

4.3.8 ReportService Class 29

4.3.8.1 Class Identification 29

4.3.8.2 Attributes (or Data members) 29

4.3.8.3 Methods (or class functions) 29

4.3.8.4 Methods (or class functions) 31

4.3.8.5 Methods (or class functions) 31

4.3.8.6 Methods (or class functions) 31

4.3.8.7 Methods (or class functions) 31

4.3.8.8 Methods (or class functions) 32

4.3.8.9 Methods (or class functions) 32

4.4 MANAGERS 32

4.4.1 BlockedCardManager Class 32

4.4.1.1 Class Identification 32

4.4.1.2 Attributes (or Data members) 33

4.4.1.3 Methods (or class functions) 33

4.4.2 CardManager Class 33

4.4.2.1 Class Identification 33

4.4.2.2 Attributes (or Data members) 34

4.4.2.3 Methods (or class functions) 34

4.4.2.4 Methods (or class functions) 34

4.4.2.5 Methods (or class functions) 35

4.4.3 CustomerManager Class 35

4.4.3.1 Class Identification 35

4.4.3.2 Attributes (or Data members) 35

4.4.3.3 Methods (or class functions) 35

4.4.4 PaymentManager Class 36

4.4.4.1 Class Identification 36

4.4.4.2 Attributes (or Data members) 36

4.4.4.3 Methods (or class functions) 36

4.4.5 SchemeManager Class 37

4.4.5.1 Class Identification 37

4.4.5.2 Attributes (or Data members) 37

4.4.5.3 Methods (or class functions) 37

4.4.6 TransactionManager Class 38

4.4.6.1 Class Identification 38

4.4.6.2 Attributes (or Data members) 39

4.4.6.3 Methods (or class functions) 39

4.4.7 LoginManager Class 40

4.4.7.1 Class Identification 40

4.4.7.2 Attributes (or Data members) 40

4.4.7.3 Methods (or class functions) 40

4.4.8 ReportManager Class 41

4.4.8.1 Class Identification 41

4.4.8.2 Attributes (or Data members) 41

4.4.8.3 Methods (or class functions) 41

4.4.8.4 Methods (or class functions) 42

4.4.8.5 Methods (or class functions) 42

4.4.8.6 Methods (or class functions) 43

4.4.8.7 Methods (or class functions) 43

4.4.8.8 Methods (or class functions) 43

4.4.8.9 Methods (or class functions) 43

4.5 Wrapper Class 44

4.5.1 InfyCreditCardWrapper 44

4.5.1.1 Class Identification 44

4.5.1.2 Attributes (or Data members) 44

4.5.1.3 Methods (or class functions) 44

4.6 MANAGED BEANS 46

4.6.1 BlockCard MB 46

4.6.1.1 Class Identification 46

4.6.1.2 Attributes (or Data members) 46

4.6.2 Payment MB 47

4.6.2.1 Class Identification 47

4.6.2.2 Attributes (or Data members) 47

4.6.3 ViewCardDetailsMB 48

4.6.3.1 Class Identification 48

4.6.3.2 Attributes (or Data members) 48

4.6.4 UnblockCardMB 49

4.6.4.1 Class Identification 49

4.6.4.2 Attributes (or Data members) 49

4.6.5 Card Usage MB 50

4.6.5.1 Class Identification 50

4.6.5.2 Attributes (or Data members) 50

4.6.6 CardApplication MB 51

4.6.6.1 Class Identification 51

4.6.6.2 Attributes (or Data members) 51

4.6.7 Login MB 53

4.6.7.1 Class Identification 53

4.6.7.2 Attributes (or Data members) 53

4.6.8 Customer MB 54

4.6.8.1 Class Identification 54

4.6.8.2 Attributes (or Data members) 54

4.6.9 Card MB 54

4.6.9.1 Class Identification 54

4.6.9.2 Attributes (or Data members) 54

4.6.10 SchemeMB 55

4.6.10.1 Class Identification 55

4.6.10.2 Attributes (or Data members) 55

4.6.11 TransactionMB 56

4.6.11.1 Class Identification 56

4.6.11.2 Attributes (or Data members) 57

4.6.12 RevenueStatisticsMB 57

4.6.12.1 Class Identification 57

4.6.12.2 Attributes (or Data members) 57

4.6.13 TransactionStatisticsMB 58

4.6.13.1 Class Identification 58

4.6.13.2 Attributes (or Data members) 58

4.6.14 BlockStatisticsMB 59

4.6.14.1 Class Identification 59

4.6.14.2 Attributes (or Data members) 60

4.6.15 SchemeReportMB 60

4.6.15.1 Class Identification 60

4.6.15.2 Attributes (or Data members) 60

4.6.16 CardStatusMB 61

4.6.16.1 Class Identification 61

4.6.16.2 Attributes (or Data members) 61

5. Summary 63

# Overview

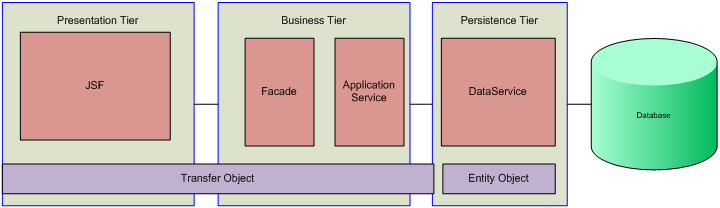
In any of the software development process the design stage is having an immense importance. In design phase the entire software gets a structure, in the sense we plan the functional and non functional aspects of the software. The software requirement specification (SRS) is the input to this stage. Based on the functional and non-functional requirements of the customer, the designers design the system. The design includes the domain, data base designs. In the domain design we determine which application program can be used to develop the software. Based on the methodology (structured, object oriented, event driven) used by the language the domain design is done. In the data base design, depending on the backend data base the suitable structure for storing the data (tables, objects) are selected.

For the ICCI Store systems the front end technology should be an object oriented language like JAVA so that we can leverage the benefits of object oriented programming. The benefits includes platform independence, robustness etc. Since we are using JAVA for the development, the required classes should be designed. The UML diagrams can be used for the class design. The UML diagrams describe the member variables, the methods that a class can have.

In the backend we are using Oracle data base for the permanent storage of data. Oracle is using the relational data model, so the tables and their relationships should be identified in this document. The tables should be designed in such a way that the high performance in the data base related activities must be ensured.

# High level design architecture

The following diagram shows the big picture of the ICCI system architecture.



The Presentation tier (which includes the JSF and JSP components) talks to the Business Tier (which consists of the Wrapper and Manager classes). The Manager classes implements the business logic and the methods in these classes are accessed by the presentation tier through Wrapper Class. The Manager classes use the Data Service classes to make communication with the database and fetch and manipulate data.



## Subsystems

The system includes following subsystems:

Customer Module:

* Login
* Apply for Card
* View Card Details
* View Card Usage
* Online Transaction
* Choose Scheme

Company Representative Module:

* Make Payment
* Block Card
* Unblock Card
* Register Card
* Transaction Statistics
* Revenue Statistics
* Blocked Card Report
* Scheme Report
* Approved & Rejected/Pending Card Report

## Modeling

Based on the Use Case model defined in the requirement engineering phase the following models need to be developed.

* Object Model
* Deployment Model

## Layering and Partitioning

This application follows MVC and layered architectural approach.

MVC architecture has 3 Components

* + Model
  + View
  + Controller

Model: Responsible for the business logic and the business data

View: Responsible for displaying the data from the Model to the Client

Controller: Responsible for the flow and state of the user input

## Categorization of classes

The classes used in ICCI are categorized as follows.

1. Wrapper Class
2. Manager Class
3. Data Service Class
4. Entity Bean Class
5. Transfer Object classes
6. Managed Bean classes

## Coding and UI standards

Coding standards as prescribed by Sun Microsystems should be followed which includes usage of camel Casing notation for variable names. Coding standard of ICCI development is also based on the Java Coding Standards.

UI should be smooth and consistent across all the pages in look and feel.

## System test planning

System test plan is not a part of this document. It has to be defined in a separate document.

## Development environment

For the development of the ICCI system the following specifications are used.

RAM: 256 MB

HDD: 40 GB

Network: Above 56 Kbps

Operating System: Windows XP SP2/Vista

## Software

Eclipse 3.2.4 Ganymede IDE with supports for JSF and JPA

Java Runtime Environment 1.6

## Hardware/software and platform

For the deployment of the ICCI system the following specifications are used.

RAM: 256 MB

HDD: 40 GB

Network: Above 100 Mbps

Operating System: Windows XP SP2/Vista

## Components off the Shelf (COTS)

All the subsystems of the enterprise application are built from the scratch. Therefore the ICCI is not using any COTS packages.

# Detailed Design

The detailed design of the system is been captured using object model and deployment module. The details of these model is been captured in the subsequent sub sections.



## Implementation elements

Every element of the subsystem is defined in detail in class diagram. Please refer the table below to know the various implementation elements of InfyCreditCardInc.

|  |  |
| --- | --- |
| ***Managed Bean Class*** | |
| BlockCardMB | CardApplicationMB |
| CardUsageMB | PaymentMB |
| TransactionMB | UnblockCardMB |
| ViewCardDetailsMB | CardMB |
| CustomerMB | SchemeMB |
| CardStatusMB | SchemeReportMB |
| RevenueStatisticsMB | BlockStatisticsMB |
| LoginMB | TransactionStatisticsMB |
| ***TO Class*** | |
| BlockedCardTO | CardApplicationTO |
| CardTO | TransactionTO |
| LoginTO | SchemeTO |
| CustomerTO | PaymentTO |
| ApprovedCardTO |  |
| ***Manager Class*** | |
| CardManager | LoginManager |
| CustomerManager | SchemeManager |
| PaymentManager | TransationManager |
| ReportManager |  |
| ***Service Class*** | |
| CardService | LoginService |
| CustomerService | TransactionService |
| PaymentService | SchemeService |
| ReportService |  |

## Design of algorithms

The system is using standard algorithms for functionalities including searching, sorting, retrieving the data, deleting the data etc. There is no special critical algorithms is used in ICCI.

## Database design

Database design is not included as a part of this document. It is documented separately in ICCI\_DBDesign.doc.

## Unit test planning

Individual units of each subsystems of ICCI need to be tested in isolation by the developer of the unit. To perform this activity unit test plan (UTP) is to be prepared based on the template taken from PRIDE.

## Optimization of algorithms and data access

The following techniques are used to optimise the database implementation.

* Proper usage of index.
* Partitioning of the customer order data has to be done.
* All the tables are normalized not more than 3NF.

## External interfaces design

Not Applicable

## Error messages

* “Mandatory Field: Please enter the value for <<Xxx>>”.
* “Numeric Field : Please enter only numbers for <<Xxx>>”.
* “Invalid Date format: Please enter Date in “DD-MON-YYYY format”

## Screen layout

Screen layout is not included as a part of this document. It is documented separately in ICCI\_Screenshot.doc document.

## Source for output fields

Source for output fields are documented in Use-Case specification document.

## Special processing notes

### Assumptions

* Payment for the order will be paid by the customer during delivery.
* Shipping is done to the customers.
* The Geographies are going to remain the same for the near future.
* The organisational roles are going to remain the same for the near future.
* All the procurements are supplied by vendors.

### Limits

* No provision for userId/Password change.

### Exception handling

The exception handling of ICCI has to be robust. Following exceptions are identified based on the captured requirements.

* CannotUnblockCard
* CardAlreadyBlocked
* CardNotBlocked
* InvalidCardNo
* InvalidCredentials
* InvalidCustomerId
* InvalidPin
* InvalidScheme
* NoBalanceToBePaid
* NoDataFoundException
* SchemeAlreadyChosen
* TableEmpty

# class design specification

## ENTITY BEANS

### BlockedCard Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | BlockedCardEntity |
| **Class Description** | This class is mapped to the ICCI\_BLOCKEDCARD. |
| **Class Inheritance** | NA |
| **Classes Referenced** | CardEntity |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| blockId | uniqueId which identifies a blocked card | int | *private* |  |
| card | A reference to CardEntity. Used to store cardNo | Long | *private* |  |
| dateOfBlock | The date on which card was blocked | Calendar | *private* |  |
| description | Reason for card block | String | *private* |  |
| status | Conveys whether a card is blocked or not | char | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Card Application Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardApplicationEntity |
| **Class Description** | This class is mapped to the ICCI\_CARD\_APPLICATION\_DETAILS. |
| **Class Inheritance** | NA |
| **Classes Referenced** | NA |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| applicationId | The application id.Generated by a sequence | int | *private* | 40001-40099 |
| name | Name of the applicant | String | *private* |  |
| email | Email of the applicant | String | *private* |  |
| phone | Phone no of the applicant | String | *private* |  |
| address | Address of the applicant | String | *private* |  |
| dateOfApplication | Date of application | Calendar | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Card Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardEntity |
| **Class Description** | This class is mapped to the ICCI\_CARD. |
| **Class Inheritance** | NA |
| **Classes Referenced** | SchemeEntity, CustomerEntity |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | The credit card number | long | *private* |  |
| pin | The credit card pin number | int | *private* |  |
| scheme | A reference to scheme entity. Used to store schemeid | schemeEntity | *private* |  |
| balanceAmount | The balance amount to be paid | double | *private* |  |
| cardAmount | The amount left in credit card | double | *private* |  |
| dateOfRegistration | Date of registration of card | Calendar | *private* |  |
| customer | A reference to customerEntity. Used to store customerId | customerEntity | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Customer Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CustomerEntity |
| **Class Description** | This class is mapped to the ICCI\_CUSTOMER. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| customerId | CustomerId of the customer | int | *private* |  |
| name | Name of the customer | String | *private* |  |
| address | Address of the customer | String | *private* |  |
| phone | Phone number of the customer | String | *private* |  |
| email | Email address of the customer | String | *private* |  |
| userName | Username of the customer (used while logging in) | String | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Payment Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | PaymentEntity |
| **Class Description** | This class is mapped to the ICCI\_PAYMENT. |
| **Class Inheritance** | NA |
| **Classes Referenced** | cardEntity |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| paymentId | Unique id for payment made | int | *private* |  |
| card | A reference to cardEntity. Used to store card number | long | *private* |  |
| paymentType | The type of payment made: full, minimum, backlog | char | *private* |  |
| amountPaid | The amount paid for payment | double | *private* |  |
| dateOfPayment | Date of Payment | Calendar | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Scheme Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | SchemeEntity |
| **Class Description** | This class is mapped to the ICCI\_SCHEME. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| schemeId | Id used to identify various schemes | char | *private* |  |
| schemeAmount | Scheme amount | double | *private* |  |
| minimumAmount | Minimum amount to be paid for choosing a particular scheme | double | *private* |  |
| interestRate | Interest rate applicable if full scheme amount is not paid | double | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Transaction Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | TransactionEntity |
| **Class Description** | This class is mapped to the ICCI\_Transaction table. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| transactionId | uniqueId which identifies the transaction on a card | int | *private* |  |
| dateOfTransaction | This is the date on which a particular transaction was made on the card | Calendar | *private* |  |
| description | this is the description on what was purchased in the transaction | String | *private* |  |
| amount | This is the amount for which the transaction was done | double | *private* |  |
| cardNo | This is the CardNO on which a transaction was made. | long | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Login Entity Bean Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | LoginEntity |
| **Class Description** | This class is mapped to the ICCI\_LOGIN table. |
| **Class Inheritance** | NA |
| **Classes Referenced** | NA |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| userId | The userId with which the user can log in | String | *Private* |  |
| password | The password with which the user can log in | String | *Private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned in section 1.1.1.2, and a default constructor are required.

## TRANSFER OBJECTS

### BlockedCardTO Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | BlockedCardTO |
| **Class Description** | The class is used to transfer blockedCard related details |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| blockId | uniqueId which identifies a blocked card | int | *private* |  |
| cardNo | Card number | long | *private* |  |
| dateOfBlock | The date on which card was blocked | Calendar | *private* |  |
| description | Reason for card block | String | *private* |  |
| status | Conveys whether a card is blocked or not | char | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### CardTO Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardTO |
| **Class Description** | The class is used to transfer card related details. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | The credit card number | long | *private* |  |
| pin | The credit card pin number | int | *private* |  |
| schemeId | SchemeId of chosen scheme | char | *private* |  |
| balanceAmount | The balance amount to be paid | double | *private* |  |
| cardAmount | The amount left in credit card | double | *private* |  |
| dateOfRegistration | Date of registration of card | Calendar | *private* |  |
| customerId | Customer Id | int | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### CardApplicationTO

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardApplicationTO |
| **Class Description** | This class is used to transfer card application & approval related data between tiers. |
| **Class Inheritance** | NA |
| **Classes Referenced** | NA |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | | **Scope** | | **Range** | |
| name | Name of the applicant | | String | | *private* | |  |
| email | Mail id of the applicant | | String | | *private* | |  |
| address | Address of the applicant | | String | | *private* | |  |
| phone | Phone no of the applicant | | String | | *private* | |  |
| applicationId | uniqueId which identifies the card application | | long | | *private* | | 40001- 40050 |
| applicationSelected | Variable to hold the status(checked or unchecked) | | boolean | | *private* | |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned in section 1.2.2.2.

### CustomerTO Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CustomerTO |
| **Class Description** | The class is used to transfer customer related details. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| customerId | CustomerId of the customer | int | *private* |  |
| name | Name of the customer | String | *private* |  |
| address | Address of the customer | String | *private* |  |
| phone | Phone number of the customer | String | *private* |  |
| email | Email address of the customer | String | *private* |  |
| userName | Username of the customer (used while logging in) | String | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Payment TO Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | PaymentTO |
| **Class Description** | The class is used to transfer payment related details. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| paymentId | Unique id for payment made | int | *private* |  |
| cardNo | Card number | long | *private* |  |
| paymentType | The type of payment made: full, minimum, backlog | char | *private* |  |
| amountPaid | The amount paid for payment | double | *private* |  |
| dateOfPayment | Date of Payment | Calendar | *private* |  |
| cardStatus | Conveys whether the card is blocked or not | String | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### SchemeTO Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | SchemeTO |
| **Class Description** | The class is used to transfer scheme related details. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| schemeId | Id used to identify various schemes | char | *private* |  |
| schemeAmount | Scheme amount | double | *private* |  |
| minimumAmount | Minimum amount to be paid for choosing a particular scheme | double | *private* |  |
| interestRate | Interest rate applicable if full scheme amount is not paid | double | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### TransactionTO

### Class Identification

|  |  |
| --- | --- |
| Class Name | TransactionTO |
| **Class Description** | This class is used to transfer transaction related data between tiers. |
| **Class Inheritance** | NA |
| **Classes Referenced** | NA |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | | **Scope** | | **Range** | |
| transactionId | uniqueId which identifies the transaction on a card | | int | | *private* | |  |
| dateOfTransaction | This is the date on which a particular transaction was made on the card | | Calendar | | *private* | |  |
| description | this is the description on what was purchased in the transaction | | String | | *private* | |  |
| amount | This is the transaction amount | | double | | *private* | | amount |
| cardNo | This is the CardNO on which a transaction was made. | | long | | *private* | | cardNo |
| balanceAmount | This is the balance amount | | double | | *private* | |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

### Login Transfer Object Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | LoginTO |
| **Class Description** | This class is used to transfer user related data between tiers. |
| **Class Inheritance** | NA |
| **Classes Referenced** | NA |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | | **Range** |
| userId | The string used for authentication to login | String | *Private* |  | |
| password | The string used for authentication to login | String | *Private* |  | |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned in section 1.1.2.2 and a default constructor are required.

### ApprovedCardTO Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | ApprovedCardTO |
| **Class Description** | The class is used to transfer card related details. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| customerName | The customer name | String | *private* |  |
| schemeAmt | The scheme amount | double | *private* |  |
| paymentType | The payment type choosen | String | *private* |  |
| dateOfBlock | The date of card blockage | Calendar | *private* |  |

### Methods (or class functions)

The appropriate Getter and Setter methods for all the Data members mentioned.

## DATA SERVICES

### BlockedCardService Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | BlockedCardService |
| **Class Description** | This class is responsible for the data service tier methods related to blocking a card |
| **Class Inheritance** | NA |
| **Classes Referenced** | BlockedCardTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public BlockedCardTO checkCardBlocked(long cardNo)** |
| Purpose and algorithm description | Check whether the card is blocked. |
| EntityBeans/ ValueObjects Used | BlockedCardTO, BlockedCardEntity |
| Exceptions | InvalidCardNo |
| Special Notes | N.A |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public BlockedCardTO blockCard(BlockedCardTO to)** |
| Purpose and algorithm description | Blocks the card whose card number is set in BlockedCardTO.  The card may be blocked because of following reasons:   * + Theft   + Payment Not done   + Other Reasons   If the amount to be paid > 2\*scheme amount of chosen scheme, block the card as Payment is not done. |
| EntityBeans/ ValueObjects Used | BlockedCardTO, BlockedCardEntity |
| Exceptions |  |
| Special Notes | N.A |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public BlockedCardTO unblockCard(BlockedCardTO to)** |
| Purpose and algorithm description | Unblocks a blockedcard whose card number is set in BlockedCardTO |
| EntityBeans/ ValueObjects Used | BlockedCardTO, BlockedCardEntity |
| Exceptions |  |
| Special Notes | N.A |

### Attributes (or Data members)

NA

### CardService Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardService |
| **Class Description** | This class is responsible for the data service tier methods related to Card |
| **Class Inheritance** | NA |
| **Classes Referenced** | CardTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public CardTO getCardDetails(CardTO to)** |
| Purpose and algorithm description | Retrieve and return the details of the card whose CardNo is specified as part of CardTO. |
| EntityBeans/ ValueObjects Used | CardTO |
| Exceptions | InvalidCardNo |
| Special Notes | N.A |

### Attributes (or Data members)

NA

|  |  |
| --- | --- |
| **Method** | **public CardTO applyCard(CardApplicationTO cardApplicationTO)** |
| Purpose and algorithm description | The method is used persist the card application details in the database. |
| EntityBeans/ ValueObjects Used | CardApplicationTO,CardApplicationEntity |
| Exceptions |  |
| Special Notes | N.A |

### Attributes (or Data members)

NA

|  |  |
| --- | --- |
| **Method** | **public List<CardApplicationTO> retrieveAppliedCardDetails()** |
| Purpose and algorithm description | The method is used to retrieve the applied card details |
| EntityBeans/ ValueObjects Used | CardApplicationTO,CardApplicationEntity |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public void approveCard(int applicationId)** |
| Purpose and algorithm description | The method retrieves the card application details corresponding to the passed application id. With the obtained details generate userId & password to populate LoginEntity. For eg: If applicant name is David userId should be of 6 characters with the first three characters of this name concatenated with 3 digit random number eg:Dav629 & password should be of at least 8 random characters. Generate a 16digit & 4 digit random numbers to set the cardNo & pin for the applicant. Populate the CustomerEntity & CardEntity.Then LoginEntity & CardEntity is persisted. After that the CardApplicationEntity corresponding to the passed applicationId is removed. |
| EntityBeans/ ValueObjects Used | CardApplicationEntity, LoginEntity, CustomerEntity, CardEntity |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

### CustomerService Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CustomerService |
| **Class Description** | This class is responsible for the data service tier methods related to Customer |
| **Class Inheritance** | NA |
| **Classes Referenced** | CustomerTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public CustomerTO getCustomeDetails(CustomerTO to)** |
| Purpose and algorithm description | Retrieve and return the details of the customer whose customerid is specified as part of CustomerTO. |
| EntityBeans/ ValueObjects Used | CustomerTO |
| Exceptions | InvalidCustomerId |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public CustomerTO getCustomerId(LoginTO loginTo)** |
| Purpose and algorithm description | Gets the customerId of the user. |
| EntityBeans/ ValueObjects Used | Customer Entity |
| Exceptions | SQLException, Exception |
| Special Notes | N.A |

### SchemeService Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | SchemeService |
| **Class Description** | This class is responsible for the data service tier methods related to available schemes. |
| **Class Inheritance** | NA |
| **Classes Referenced** | SchemeTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public SchemeTO getSchemeDetails(SchemeTO to)** |
| Purpose and algorithm description | Retrieve and return scheme details of the scheme whose schemeid is specified as part of SchemeTO |
| EntityBeans/ ValueObjects Used | SchemeTO |
| Exceptions | InvalidScheme |
| Special Notes | N.A |

### Attributes (or Data members)

NA

### 4.3.3.5Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public Boolean schemeChosen(CustomerTO customerTo)** |
| Purpose and algorithm description | Checks whether the customer has already chosen one scheme. |
| EntityBeans/ ValueObjects Used | Customer Entity, Card Entity |
| Exceptions | SQLException, Exception |
| Special Notes | N.A |

### Attributes (or Data members)

NA

### 4.3.3.7 Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public void updateSchemeDetails(CardTO cardTo)** |
| Purpose and algorithm description | Updates the card entity with the scheme details chosen by the customer. |
| EntityBeans/ ValueObjects Used | Card Entity |
| Exceptions | SQLException, Exception |
| Special Notes | N.A |

### 4.3.3.8Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public Boolean schemeChosen(CustomerTO customerTo)** |
| Purpose and algorithm description | Checks whether the customer has already chosen one scheme. |
| EntityBeans/ ValueObjects Used | Customer Entity, Card Entity |
| Exceptions | SQLException, Exception |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public void updateSchemeDetails(CardTO cardTo)** |
| Purpose and algorithm description | Updates the card entity with the scheme details chosen by the customer. |
| EntityBeans/ ValueObjects Used | Card Entity |
| Exceptions | SQLException, Exception |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public CardTO getCardDetailsForUpdate (CardTO cardTO)** |
| Purpose and algorithm description | * Get the card details for the particular customer |
| EntityBeans/ ValueObjects Used | CardEntity |
| Exceptions |  |
| Special Notes | N.A |

### PaymentService Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | PaymentService |
| **Class Description** | This class is responsible for the data service tier methods related to payment |
| **Class Inheritance** | NA |
| **Classes Referenced** | PaymentTO, CardTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public PaymentTO getPaymentDetails(CardTO to)** |
| Purpose and algorithm description | Retrieve and return payment details related to a cardNo like lastPaymentDate, type of payment made, amount paid etc |
| EntityBeans/ ValueObjects Used | PaymentTO, CardTO |
| Exceptions | InvalidCardNo, ParseException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public PaymentTO makePayment(PaymentTO to)** |
| Purpose and algorithm description | Retrieve and return payment details after payment is made |
| EntityBeans/ ValueObjects Used | PaymentTO |
| Exceptions | InvalidCardNo |
| Special Notes | N.A |

### Transaction Service Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | TransactionService |
| **Class Description** | This class is responsible for the data service tier methods related to making a transaction |
| **Class Inheritance** | NA |
| **Classes Referenced** | TransactionTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public double getBalance(long cardNo)** |
| Purpose and algorithm description | return the current available cardAmount for the corresponding card. |
| EntityBeans/ ValueObjects Used | transactionTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public TransactionTO saveTransaction(TransactionTO transactionTO)** |
| Purpose and algorithm description | to persist the transaction details and to return the other transaction details like dateOfTransaction, auto generated transactionid, etc. |
| EntityBeans/ ValueObjects Used | transactionTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public void updateCardAmount(double amount, long cardNo)** |
| Purpose and algorithm description | to update the cardAmount by the specified value for the corresponding card. |
| EntityBeans/ ValueObjects Used | transactionTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<TransactionTO> cardUsage(String month, String year, long cardNo )** |
| Purpose and algorithm description | based on what month the user passes and the year (whether “current” or “previous”) return a list of TransactionTO, for the transactions **on that** **month and year chosen.** |
| EntityBeans/ ValueObjects Used | TransactionTO |
| Exceptions | NoProductsAvailableException |
| Special Notes | N.A |

### Login Service Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | LoginService |
| **Class Description** | This class is responsible for the data service methods related to login authentication and validation of the actor. |
| **Class Inheritance** | NA |
| **Classes Referenced** | LoginEntity, LoginTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public LoginTO validateLogin(LoginTO loginTO)** |
| Purpose and algorithm description | Validate the userId and password from the ICCI\_Login table. If valid user, then populate the LoginTO object with the values from the Login Entity.Then return the populated LoginTO , else throw an exception. |
| EntityBeans/ ValueObjects Used | LoginEntity,LoginTO |
| Exceptions | InvalidCredentialsException, SQLException, Exception |
| Special Notes | N.A |

### ReportService Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | ReportService |
| **Class Description** | This class is responsible for the data service tier methods related to Report generation |
| **Class Inheritance** | NA |
| **Classes Referenced** | TransactionTO |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List getRevenueDetails(Calendar fromDate, Calendar toDate)** |
| Purpose and algorithm description | * Retrieve the **sum of Payments** between the said dates from **Payment** **table** and store it as the **revenue**. * Retrieve the **Cards** which were **registered** between the said dates. * Using those card objects calculate the **sum of SchemeAmounts** corresponding to those cards, from the **scheme** **Table**. * Store the value as **expenditure**. * Store **revenue-expenditure** as **profit**. * Place all the values in a **list** and **return** the list of values. |
| EntityBeans/ ValueObjects Used |  |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public Map TransactionDetails(Calendar fromDate, Calendar toDate)** |
| Purpose and algorithm description | * The card number and count of transactions carried out using the card during a specified duration are populated in a map (Key is card number and value is count of transactions) and returned |
| EntityBeans/ ValueObjects Used | TransactionEntity |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<TransactionTO> getTransactions(long cardNo, Calendar fromDate, Calendar toDate)** |
| Purpose and algorithm description | * Retrieve and return a list of the transactions carried out by specified card number during the specified duration. |
| EntityBeans/ ValueObjects Used | TransactionEntity, TransactionTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<BlockedCardTO> getBlockedDetails(String[] reasons)** |
| Purpose and algorithm description | * Retrieve and return a list of all cards blocked because of the specified reasons |
| EntityBeans/ ValueObjects Used | BlockedCardEntity, BlockedCardTO |
| Exceptions |  |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public List<CardTO> getCustomersForScheme(SchemeTO schemeTo) throws SchemeNotChosenException** |
| Purpose and algorithm description | This method gets the card details for the particular scheme. If the scheme is not chosen , then it throws SchemeNotChosen Exception. |
| EntityBeans/ ValueObjects Used |  |
| Exceptions | SchemeNotChosen |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public CustomerTO getCustomersDetails(CustomerTO customerTo)** |
| Purpose and algorithm description | This method gets the details of the particular customer. |
| EntityBeans/ ValueObjects Used |  |
| Exceptions |  |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List<CardApplicationTO> CardRejectedDetails(Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | The method retrieves all the card application details between the fromDate & toDate to obtain the pending or rejected card application details. |
| EntityBeans/ ValueObjects Used | CardApplicationTO,CardApplicationEntity |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List<ApprovedCardTO> CardApprovedDetails(Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | The method retrieves the Customer name, Scheme amount, Payment type, DateOfBlock between the fromDate & toDate , to populate the ApprovedCardTO object. (if any of the property is null set as NA) |
| EntityBeans/ ValueObjects Used | CardEntity,SchemeEntity,PaymentEntity,CustomerEntity,ApprovedCardTO |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public Map<String,Integer> statusDetails(Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | The method retrieves the count of card approved & card application rejected/pending between the given dates. With the details sets the map & returns it. |
| EntityBeans/ ValueObjects Used | CardApplicationEntity, CardEntity |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List getCardNumbers(Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | The method retrieves the cardNo’s between the given dates & returns it. |
| EntityBeans/ ValueObjects Used | CardEntity |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

## MANAGERS

### BlockedCardManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | BlockedCardManager |
| **Class Description** | This class is responsible for interaction with the service class |
| **Class Inheritance** | NA |
| **Classes Referenced** | BlockedCardTO, BlockedCardService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public BlockedCardTO checkCardBlocked(BlockedCardTO to)** |
| Purpose and algorithm description | * Checks whether specified card is blocked. * If not blocked, the card is blocked |
| EntityBeans/ ValueObjects Used | BlockedCardTO, BlockedCardService |
| Exception | InvalidCardNo, CardAlreadyBlocked |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public BlockedCardTO unblockCard(BlockedCardTO to)** |
| Purpose and algorithm description | * If the specified card is blocked, unblock the card. * The card may be blocked because of following reasons:   + Theft   + Payment Not done   + Other Reasons   Unblock the card it the payment is done if the reason for blocking the card is “Payment Not Done”. |
| EntityBeans/ ValueObjects Used | BlockedCardTO, BlockedCardService |
| Exceptions | InvalidCardNo, CardAlreadyBlocked, CardNotBlocked, CannotUnblockCard |
| Special Notes | N.A |

### CardManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardManager |
| **Class Description** | This class is responsible for interaction with the service class |
| **Class Inheritance** | NA |
| **Classes Referenced** | CardTO, CardService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public CardTO getCardDetails(CardTO to)** |
| Purpose and algorithm description | * Invokes the appropriate service class method to get the details of the card whose card number is set in CardTO. |
| EntityBeans/ ValueObjects Used | CardTO, CardService |
| Exception | InvalidCardNo |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public CardTO getCardDetailsForLogin(LoginTO loginTO)** |
| Purpose and algorithm description | * Gets the card details for the card and checks for the pin. |
| EntityBeans/ ValueObjects Used | CardTO, CardService |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public CardTO applyCard(CardApplicationTO cardApplicationTO)** |
| Purpose and algorithm description | * Creates an object of CardService class * Calls the applyCard() method of CardService class by passing the cardApplicationTO object |
| EntityBeans/ ValueObjects Used | CardService |
| Exceptions |  |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public void approveCard(int applicationId)** |
| Purpose and algorithm description | * Creates an object of CardService class * Calls the approveCard() method of CardService class by passing the applicationId |
| EntityBeans/ ValueObjects Used | CardService |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List<CardApplicationTO> retrieveAppliedCardDetails()** |
| Purpose and algorithm description | * Creates an object of CardService class * Calls the retrieveAppliedCardDetails() method of CardService class & retrieves the CardApplicationTO objects |
| EntityBeans/ ValueObjects Used | CardService |
| Exceptions | NoDataFoundException |
| Special Notes | N.A |

### CustomerManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | CustomerManager |
| **Class Description** | This class is responsible for interaction with the service class |
| **Class Inheritance** | NA |
| **Classes Referenced** | CustomerTO, CustomerService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public CustomerTO getCustomerDetails(CustomerTO to)** |
| Purpose and algorithm description | * Invokes the appropriate service class method to get the details of the customer whose customerid is set in CustomerTO. |
| EntityBeans/ ValueObjects Used | CustomerTO, CustomerService |
| Exception | InvalidCustomerId |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public CustomerTO getCustomerId(LoginTO loginTO)** |
| Purpose and algorithm description | * Gets the customer details for the user. |
| EntityBeans/ ValueObjects Used | LoginTO,CustomerTO, CustomerService |
| Exceptions |  |
| Special Notes | N.A |

### PaymentManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | PaymentManager |
| **Class Description** | This class is responsible for interaction with the service class |
| **Class Inheritance** | NA |
| **Classes Referenced** | PaymentTO, CardTO, PaymentService, BlockedCardTO, BlockedService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public PaymentTO getPaymentDetails(CardTO to)** |
| Purpose and algorithm description | * Checks whether specified card is blocked. * The details are populated and returned in PaymentTO. |
| EntityBeans/ ValueObjects Used | PaymentTO, PaymentService |
| Exception | InvalidCardNo, ParseException |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public PaymentTO makePayment(PaymentTO to)** |
| Purpose and algorithm description | * The final balance amount to be paid is calculated. * If the card is blocked, it is checked if the payment made causes the card to be unblocked. * The payment details are populated in PaymentTO and passed on to appropriate service class method for persistence. |
| EntityBeans/ ValueObjects Used | CardTO, SchemeTO, PaymentTO, PaymentService |
| Exceptions | InvalidCardNo, InvalidScheme |
| Special Notes | N.A |

### SchemeManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | SchemeManager |
| **Class Description** | This class is responsible for interaction with the service class |
| **Class Inheritance** | NA |
| **Classes Referenced** | SchemeTO, SchemeService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public SchemeTO getSchemeDetails(SchemeTO to)** |
| Purpose and algorithm description | * Invokes the appropriate service class method to get the details of the scheme whose schemeid is set in SchemeTO. |
| EntityBeans/ ValueObjects Used | SchemeTO, SchemeService |
| Exception | InvalidCardNo, InvalidScheme |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public SchemeTO getSchemeDetails(SchemeTO schemeTO)** |
| Purpose and algorithm description | * Gets the scheme details for a particular scheme by calling the service method |
| EntityBeans/ ValueObjects Used | SchemeTO,SchemeService |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public boolean schemeChosen(CustomerTO customerTO)** |
| Purpose and algorithm description | * Checks if the customer has already chosen a scheme. |
| EntityBeans/ ValueObjects Used | CustomerTO,SchemeService |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public void upDateSchemeDetails(CardTO cardTO)** |
| Purpose and algorithm description | * Updates the scheme details by calling the appropriate service method. |
| EntityBeans/ ValueObjects Used | CardTO, SchemeService |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **Public CardTO getCardDetailsForUpdate (CardTO cardTO)** |
| Purpose and algorithm description | * Get the card details for the customer by calling the service method. |
| EntityBeans/ ValueObjects Used | CardTO, SchemeService |
| Exceptions |  |
| Special Notes | N.A |

### TransactionManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | TransactionManager |
| **Class Description** | This class is responsible for the business service tier methods related to Transaction |
| **Class Inheritance** | NA |
| **Classes Referenced** | TransactionTO, TransactionService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public TransactionTO saveTransaction(TransactionTO transactionTO)** |
| Purpose and algorithm description | * based on session handling retrieve the cardNo used for logging in. * pass the card no to getBalance() method of TransactionService * If the balance returned is less than the transaction Amount raise an exception. * If balance is greater than transaction amount, call the updatecardAmount() method of TransactionService * After that call the saveTransaction() method of TransactionService. |
| EntityBeans/ ValueObjects Used | TransactionTO, TransactionService |
| Exceptions | lowBalanceException |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List getMonthList()** |
| Purpose and algorithm description | * This method is used to generate a **list of list of months** to be used to populate the drop down in front end. * This List is used to populate the drop down in front end, which displays a list of last 6 months from the current date. * Please refer the diagram for further clarification. * The purpose is to make sure that only the last 6 months are displayed depending on the current date. * List 1 contains the months that fall in the current year and List 2 contains months that fall in the previous year. * Add both List 1 and List 2 to a List and return it. |
| EntityBeans/ ValueObjects Used | TransactionTO, TransactionService |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<TransactionTO> cardUsage(String month, String year)** |
| Purpose and algorithm description | * This method uses the month and year selected by the user and returns a List of TransactionTO Objects. * Month is the ‘month’ chosen from drop down. * Year is the ‘year’ chosen from the radio button at front end. * In this method, the card No is obtained using session handling. Using sessions, obtain the card no which the user used to login. * Call the cardUsage() of TransactionService by passing the month, year and cardNo. |
| EntityBeans/ ValueObjects Used | TransactionTO, TransactionService |
| Exceptions |  |
| Special Notes | N.A |

### LoginManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | LoginManager |
| **Class Description** | This class is responsible for interaction with the service class |
| **Class Inheritance** | NA |
| **Classes Referenced** | LoginTO, LoginService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public LoginTO validateLogin(LoginTO to)** |
| Purpose and algorithm description | * Gets the details for the particular user. * Checks if the password entered is valid. |
| EntityBeans/ ValueObjects Used | LoginTO,LoginService |
| Exception | InvalidCredentials |
| Special Notes | N.A |

### ReportManager Class

### Class Identification

|  |  |
| --- | --- |
| Class Name | ReportManager |
| **Class Description** | This class is responsible for interaction with the manager class for report generation |
| **Class Inheritance** | NA |
| **Classes Referenced** | ReportService |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public Map TransactionDetails(Calendar fromDate, Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **TransactionDetails()** method by passing the parameters. |
| EntityBeans/ ValueObjects Used | ReportService |
| Exception | TableEmpty |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<TransactionTO> getTransactions(long cardNo, Calendar fromDate, Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **getTransactions ()** method by passing the parameters. |
| EntityBeans/ ValueObjects Used | ReportService |
| Exception |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<BlockedCardTO> getBlockedDetails(String[] reasons)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **getBlockedDetails ()** method by passing the parameters. |
| EntityBeans/ ValueObjects Used | ReportService |
| Exception | NoCardsBlockedForReason |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List getRevenueDetails(Calendar fromDate, Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **getRevenueDetails()** method by passing the parameters. |
| EntityBeans/ ValueObjects Used | LoginTO,LoginService |
| Exception | InvalidCredentials |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public List<CardTO> getCustomersForScheme(SchemeTO schemeTo) throws SchemeNotChosenException** |
| Purpose and algorithm description | This method gets the card details for the particular scheme by invoking the corresponding service method. If the scheme is not chosen , then it throws SchemeNotChosen Exception. |
| EntityBeans/ ValueObjects Used |  |
| Exception | SchemeNotChosen |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **Public CustomerTO getCustomerDetails(CustomerTO customerTo)** |
| Purpose and algorithm description | This method gets the details of the particular customer by invoking the corresponding service method. Invokes the getCustomerDetails and getCardDetailsForUpdate methods of the wrapper. |
| EntityBeans/ ValueObjects Used |  |
| Exceptions |  |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List getCardNumbers(Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **getCardNumbers**() method by passing the parameters. |
| EntityBeans/ ValueObjects Used | **ReportService** |
| Exception | NoDataFoundException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List<CardApplicationTO> CardRejectedDetails (Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **CardRejectedDetails** () method by passing the parameters. |
| EntityBeans/ ValueObjects Used | **ReportService** |
| Exception | NoDataFoundException |
| Special Notes | N.A |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public List<ApprovedCardTO> CardApprovedDetails (Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **CardApprovedDetails** () method by passing the parameters. |
| EntityBeans/ ValueObjects Used | **ReportService** |
| Exception | NoDataFoundException |

### Methods (or class functions)

|  |  |
| --- | --- |
| **Method** | **public Map<String,Integer> statusDetails( Calendar fromDate,Calendar toDate)** |
| Purpose and algorithm description | * Creates an Object of **ReportService** and invokes the **statusDetails** () method by passing the parameters. |
| EntityBeans/ ValueObjects Used | **ReportService** |
| Exception | NoDataFoundException |

## Wrapper Class

### InfyCreditCardWrapper

### Class Identification

|  |  |
| --- | --- |
| Class Name | InfyCreditCardWrapper |
| **Class Description** | This class is responsible for acting as an interface between the Manager tier and the controller servlet in the presentation tier. |
| **Class Inheritance** | NA |
| **Classes Referenced** | All TO’s and All BS classes |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

NA

### Methods (or class functions)

All the below mentioned methods should to call their respective methods in the respective Manager classes. The methods in both the Wrapper class and the Managers classes are the same. No logic is implemented in the method in this class.

|  |  |
| --- | --- |
| **Method** | **public TransactionTO saveTransaction(TransactionTO to)** |

|  |  |
| --- | --- |
| **Method** | **public List<TransactionTO> cardUsage(String month, String option)** |

|  |  |
| --- | --- |
| **Method** | **public CardApplicationTO applyCard(CardApplicationTO to)** |

|  |  |
| --- | --- |
| **Method** | **public List<CardApplicationTO> retrieveAppliedCardDetails()** |

|  |  |
| --- | --- |
| **Method** | **public void approveCard(int applicantId)** |

|  |  |
| --- | --- |
| **Method** | **public PaymentTO getPaymentDetails(CardTO to)** |

|  |  |
| --- | --- |
| **Method** | **public PaymentTO makePayment(PaymentTO to)** |

|  |  |
| --- | --- |
| **Method** | **public BlockedCardTO checkCardBlocked(BlockedCardTO to)** |

|  |  |
| --- | --- |
| **Method** | **public CardTO getCardDetails(CardTO to)** |

|  |  |
| --- | --- |
| **Method** | **public SchemeTO getSchemeDetails(SchemeTO to)** |

|  |  |
| --- | --- |
| **Method** | **public CustomerTO getCustomerDetails(CustomerTO to)** |

|  |  |
| --- | --- |
| **Method** | **public BlockedCardTO unblockCard(BlockedCardTO to)** |

|  |  |
| --- | --- |
| **Method** | **public LoginTO validateLogin(LoginTO to)** |

|  |  |
| --- | --- |
| **Method** | **public CardTO getCardDetailsForLogin(CardTO to)** |

|  |  |
| --- | --- |
| **Method** | **public CustomerTO getCustomerId(LoginTO to)** |

|  |  |
| --- | --- |
| **Method** | **public boolean schemeChosen(CustomerTO customerTo)** |

|  |  |
| --- | --- |
| **Method** | **public void upDateSchemeDetails(CardTO cardTo)** |

|  |  |
| --- | --- |
| **Method** | **public CardTO getSchemeDetails(CardTO cardTo)** |

|  |  |
| --- | --- |
| **Method** | **public Map TransactionDetails(Calendar fromDate, Calendar toDate)** |

|  |  |
| --- | --- |
| **Method** | **public List<TransactionTO> getTransactions(long cardNo, Calendar fromDate, Calendar toDate)** |

|  |  |
| --- | --- |
| **Method** | **public List<BlockedCardTO> getBlockedDetails(String[] reasons)** |

|  |  |
| --- | --- |
| **Method** | **public List getRevenueDetails(Calendar fromDate, Calendar toDate)** |

|  |  |
| --- | --- |
| **Method** | **Public List<CardTO> getCustomersForScheme(SchemeTO schemeTo)** |

|  |  |
| --- | --- |
| **Method** | **Public CustomerTO getCustomersDetails(CustomerTO customerTo)** |

|  |  |
| --- | --- |
| **Method** | **public List getCardNumbers(Calendar fromDate,Calendar toDate)** |
| **Method** | **public List<CardApplicationTO> CardRejectedDetails (Calendar fromDate,Calendar toDate)** |
| **Method** | **public List<ApprovedCardTO> CardApprovedDetails (Calendar fromDate,Calendar toDate)** |
| **Method** | **public Map<String,Integer> statusDetails (Calendar fromDate,Calendar toDate)** |

## MANAGED BEANS

### BlockCard MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | BlockCardMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | Used to store the 4 parts of 16-digit card number accepted from the user as a group of 4 digits. | String[] | *private* |  |
| cause | Reason for the blocking of card | String | *private* |  |
| disable | Used to enable or disable the ‘causes’ textarea filed in the JSP. The textarea is enabled if the cause of card blockage is ‘others’. | boolean | *private* |  |
| causes | The List containing the possible reasons for the blocking of the card | List<SelectItem> | *private* |  |
| message | Used to display messages to end user | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String blockCard()** |
| Purpose and algorithm description | * This method is the action listener for the submit button. * It blocks the card whose card number is input by the user |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, BlockedCardTO |
| Exceptions |  |
| Special Notes | N.A |

### Payment MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | PaymentMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | Used to store the 4 parts of 16-digit card number accepted from the user as a group of 4 digits. | String[] | *private* |  |
| paymentType | Type of payment done: Full, Minimum, Backlog | Character | *private* |  |
| customerId | Customer Id | Integer | *private* |  |
| customerName | Name of the customer | String | *private* |  |
| balanceAmount | The balance amount to be paid | Double | *private* |  |
| lastPaymentDate | The date of last payment | Calendar | *private* |  |
| cardStatus | Whether the card is blocked or not | String | *private* |  |
| disable | It is used to enable or disable the ‘makePayment’ button. The commandButton is disabled if no payment is to be done | Boolean | *private* |  |
| message | Used to display messages to users | String | *private* |  |
| amountPaid | Amount paid by the customer | Double | *private* |  |
| balance | The balance amount left to be paid by the customer | Double | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String getPaymentInformation** |
| Purpose and algorithm description | * Used to retrieve the payment details of the entered card number |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, CardTO, PaymentTO, CustomerTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String calculateBalance** |
| Purpose and algorithm description | * Used to calculate the balance amount after payment |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, CardTO, SchemeTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String makePayment()** |
| Purpose and algorithm description | * The action listener used to make payment |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, PaymentTO |
| Exceptions |  |
| Special Notes | N.A |

### ViewCardDetailsMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | ViewCardDetailsMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | Card number taken from the session object | Long | *private* |  |
| customerName | Name of the customer | String | *private* |  |
| balanceAmount | The balance amount to be paid | Double | *private* |  |
| lastPaymentDate | The date of last payment | Calendar | *private* |  |
| cardStatus | Whether the card is blocked or not | String | *private* |  |
| schemeId | SchemeId of chosen scheme | Character | *private* |  |
| message | Used to display messages to users | String | *private* |  |
| schemeAmount | The amount of chosen scheme | Double | *private* |  |
| cardAmount | The amount available with the card | Double | *private* |  |
| rateOfInterest | The rate at which interest is to be paid in case full scheme amount is not paid | Double | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String getCardDetails()** |
| Purpose and algorithm description | * To retrieve the details of the card whose card number is obtained from session object |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, SchemeTO, PaymentTO, CustomerTO |
| Exceptions |  |
| Special Notes | N.A |

### UnblockCardMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | UnblockCardMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | Used to store the 4 parts of 16-digit card number accepted from the user as a group of 4 digits. | String[] | *private* |  |
| message | Used to display messages to users | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String unblockCard()** |
| Purpose and algorithm description | * Used to unblock the card whose card number is entered by user |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, BlockedCardTO |
| Exceptions |  |
| Special Notes | N.A |

### Card Usage MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardUsageMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| transactionList | This is the list used to populate the data table | List<TransactionTO> | *private* |  |
| monthList | This is list of months used to populate the drop down. | List<SelectItem> | *private* |  |
| monthSelectedValue | This is the value chosen by theuser from the drop down. | String | *private* |  |
| Option | This is the valus chosen by the user from radio button, to indicate whther he wants month list for current year or previous year. | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String populateTable()** |
| Purpose and algorithm description | * This method is used as the action listener for the submit button. * This method creates a object of Wrapper and invokes the cardUsage() method of the wrapper by passing the **monthSelectedValue** and **option** chosen by the user. **Option** will either be ‘**current’ or ‘previous’.** * It returns ‘Success’ or ‘Failure’ depending on the return value of the method invoked. * On successful completion, the datatable in the front end is populated with the list of Transaction Objects returned from the cardUsage() method invocation. * Else, redirect to appropriate error page. |
| EntityBeans/ ValueObjects Used | InfyCreditaCardWrapper, TransactionTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public void populateMonths(ValueChangeEvent event)** |
| Purpose and algorithm description | * This method is used as a value change event handler for the option button. * An object of TransactionManager is created and the **getMonthList()**  methos is invoked. * If the “**current”** option is chosen then, the first List is displayed. If the “**previous”** option is chosen then the Second List is retrieved from the getMonthList() method invocation and displayed. |
| EntityBeans/ ValueObjects Used | TransactionTO, TransactionManager |
| Exceptions |  |
| Special Notes | N.A |

### CardApplication MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardApplicationMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| name | Name of the applicant | String | *private* |  |
| email | Email of the applicant | String | *private* |  |
| address | Address of the applicant | String | *private* |  |
| phone | Phone of the applicant | String | *private* |  |
| applicationId | uniqueId which identifies the card application | Integer | *private* | 40001- 40050 |
| randomCharacter | Variable to hold the random character generated | String | *private* |  |
| enteredCharacters | Variable to hold the entered character generated | String | *private* |  |
| cardTOList | This is the list used to populate the data table | List<CardApplicationTO> |  |  |
| message | Variable to hold the message | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String getRandomCharacter()** |
| Purpose and algorithm description | * This method is used for generating random characters * Implements the logic for generating random characters * It returns the random characters generated. |
| EntityBeans/ ValueObjects Used | NA |
| Exceptions | NA |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String applyCard()** |
| Purpose and algorithm description | * This method is used as a value change event handler for the Apply button. * Object of CardApplicationTO created & it is populated with the entered values. * An object of InfyCreditCardWrapper is created and the **applyCard()**  method is invoked by passing the populated CardApplicationTO object . |
| EntityBeans/ ValueObjects Used | CardApplicationTO, InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String approveCard()** |
| Purpose and algorithm description | * This method is used as a value change event handler for the Approve button. * Then iterate through the cardTOList to get the selected CardApplication details. * An object of InfyCreditCardWrapper is created and the **approveCard()**  method is invoked by passing applicationId(retrieved from CardApplicationTO object ) . |
| EntityBeans/ ValueObjects Used | CardApplicationTO, InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public List<CardApplicationTO> getCardTOList()** |
| Purpose and algorithm description | * The method populates cardTOList. * An object of InfyCreditCardWrapper is created and the retrieveAppliedCardDetails() method is called to retrieve cardApplicationTO details. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

### Login MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | LoginMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| Username | Username entered during login | String | *private* |  |
| Password | Password entered during login | String | *private* |  |
| Message | Used to display messages to the user. | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String validateLogin()** |
| Purpose and algorithm description | Used to valdate the login details entered by the user. Sets the username in session scope. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, LoginTO |
| Exceptions |  |
| Special Notes | N.A |

### Customer MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | CustomerMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| customerId | customerId entered during login | String | *private* |  |
| Username | UserName got from the session attribute set during login | String | *private* |  |
| randomCode | Stores the randomly generated alphanumeric characters | String | *private* |  |
| enteredCode | Random code entered by the user | String | *private* |  |
| Message | Displays the messages to the user | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String validateCustomer()** |
| Purpose and algorithm description | Checks if the entered code is same as the random code generated. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

### Card MB

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | The 16-digit card number got by the four card numbers. | Long | *private* |  |
| cardNo1 | 4 digit card number | String | *Private* |  |
| cardNo2 | 4 digit card number | String | *Private* |  |
| cardNo3 | 4 digit card number | String | *Private* |  |
| cardNo4 | 4 digit card number | String | *Private* |  |
| Pin | Pin Number of the card | String | *Private* |  |
| dateOfRegistration | The date on which the customer registered | Date | *Private* |  |
| cardAmount | Amount available with the card | double | *Private* |  |
| balanceAmount | Amount to be paid by the customer | double | *Private* |  |
| Message | Displays the messages to the user | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String checkCardDetails()** |
| Purpose and algorithm description | Used to check the cardNo and the pin entered by the user. If valid,sets the cardNo in session scope.It then checks if the customer has chosen a scheme by invoking schemechosen method. If the customer has not yet chosen a scheme, directs him to choose the scheme else directs to the customer home page. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, CardTO |
| Exceptions |  |
| Special Notes | N.A |

### SchemeMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | SchemeMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| cardNo | The 16-digit card number got by the four card numbers. | Long | *private* |  |
| cardNo1 | 4 digit card number | String | *Private* |  |
| cardNo2 | 4 digit card number | String | *Private* |  |
| cardNo3 | 4 digit card number | String | *Private* |  |
| cardNo4 | 4 digit card number | String | *Private* |  |
| Pin | Pin Number of the card | String | *Private* |  |
| dateOfRegistration | The date on which the customer registered | Date | *Private* |  |
| cardAmount | Amount available with the card | double | *Private* |  |
| balanceAmount | Amount to be paid by the customer | double | *Private* |  |
| Message | Displays the messages to the user | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String update()** |
| Purpose and algorithm description | Updates the scheme details for the customer if he does not have any scheme for his card. It first invokes the schemeChosen() method and if no scheme is chosen, updates the scheme details. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, CardTO,CustomerTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public void display(ValueChangeEvent event)** |
| Purpose and algorithm description | Displays the details of a corresponding scheme. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper,SchemeTO |
| Exceptions |  |
| Special Notes | N.A |

### TransactionMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | TransactionMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | | **Type** | | **Scope** | | **Range** |
| **Attribute Name** | **Description** | **Type** | | **Scope** | | **Range** | |
| transactionId | uniqueId which identifies the transaction on a card | | int | | *private* | |  |
| dateOfTransaction | This is the date on which a particular transaction was made on the card | | Calendar | | *private* | |  |
| description | this is the description on what was purchased in the transaction | | String | | *private* | |  |
| amount | This is the transaction amount | | double | | *private* | |  |
| cardNo | This is the CardNO on which a transaction was made. | | long | | *private* | |  |
| balanceAmount | This is the balance amount | | double | | *private* | |  |
| errorMessage | Displays the messages to the user | | String | | *private* | |  |

|  |  |
| --- | --- |
| **Method** | **public String saveTransaction()** |
| Purpose and algorithm description | Gets the user input and passes the TransactionTO in the saveTransaction() method invocation of the Wrapper |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, TransactionTO |
| Exceptions |  |
| Special Notes | N.A |

### RevenueStatisticsMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | RevenueStatisticsMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| fromDate | Date from which the report has to be generated | Calendar | *Private* |  |
| toDate | Date till which the report has to be generated for. | Calendar | *Private* |  |
| profit | This is the profit value of the company | double | *Private* |  |
| expenditure | This is the expenditure for the company | double | *Private* |  |
| revenue | This is the revenue earned by the company | double | *Private* |  |

|  |  |
| --- | --- |
| **Method** | **public String getRevenueDetails()** |
| Purpose and algorithm description | Gets the finance details of the company. Create a Object of InfyCreditCardWrapper and invoke the **getRevenueDetails()** by passing the **fromDate** and **toDate** as parameters.  the method invocation returns a List of financial details like profit, expenditure, revenue, etc. Display the details using a **PanelGrid** |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

### TransactionStatisticsMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | TransactionMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| custId | Unique id used to identify a customer | Integer | *private* |  |
| cardNo | Unique id used to identify a card | Long | *private* |  |
| fromDate | Start date for the duration | Calendar | *private* |  |
| toDate | End date for the duration | Calendar | *private* |  |
| customerDetails | Used to store the details of a customer | CustomerTO | *private* |  |
| customers | List of customer objects | List<CustomerTO> | *private* |  |
| count | List contains the count of transactions | List<Integer> | *private* |  |
| transaction | List of transactions | List<TransactionTO> | *private* |  |
| message | Displays the messages to the user | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public String getCustomer()** |
| Purpose and algorithm description | From the list of customers (member variable: customers), the appropriate customer details are selected based on the customerid and assigned to customerDetails member variable. |
| EntityBeans/ ValueObjects Used |  |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String getStatistics()** |
| Purpose and algorithm description | Invoke the TransactionDetails() method to get the card numbers and the count of transactions. From the retrieved card numbers, obtain and populate customerTO objects into customers member variable. Populate the count of transactions into count member variable. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper, CardTO, CustomerTO |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String getTransactions()** |
| Purpose and algorithm description | Invoke getTransactions() method of wrapper. Should populate the member variable, transaction, with the list of transactions for a particular card number and for a specified duration. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

### BlockStatisticsMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | BlockStatisticsMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| Reasons | stores selected reasons for blocking card | String[] | *private* |  |
| blockedCards | List of blockedCards | List<BlockedCardTO> | *private* |  |
| cardNo | Unique id used to identify a card | Long | *private* |  |
| cardTo | Used to store card details | CardTO | *private* |  |
| message | Displays the messages to the user | String | *private* |  |

|  |  |
| --- | --- |
| **Method** | **public void getBlockedDetails(ValueChangeEvent event)** |
| Purpose and algorithm description | This value change listener is invoked when a reason for blockage of card is selected by the user |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String getCardDetails()** |
| Purpose and algorithm description | It is used to retrieve card details of the selected card number |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

### SchemeReportMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | SchemeReportMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| customerIdList | The list of customerId | List<SelectItem> | *Private* |  |
| cardNoList | The list of cardNo | List<SelectItem> | *Private* |  |
| customerTo | The reference to CustomerTO object | CustomerTO | *Private* |  |
| cardTo | The reference to CardTOObject | CardTO | *Private* |  |
| message | Property used to display messages. | String | *Private* |  |

|  |  |
| --- | --- |
| **Method** | **public void getCustomersForScheme(ValueChangeEvent event)** |
| Purpose and algorithm description | Gets the customerIds and CardNos for a particular scheme. The event’s value would be schemeId. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| Method | **public void getCustomerandCardDetails(ValueChangeEvent event)** |
| Purpose and algorithm description | Gets the customer details and card details for the customerId and cardNo passed respectively. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

### CardStatusMB

### Class Identification

|  |  |
| --- | --- |
| Class Name | CardStatusMB |
| **Class Description** | This is used as the backing bean for the JSP. |
| **Class Inheritance** | NA |
| **Classes Referenced** |  |
| **Sub System** | NA |
| **Class Type** | *Complete* |
| **Change History** | *NA* |

### Attributes (or Data members)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Type** | **Scope** | **Range** |
| fromDate | Date from which the report has to be generated | Calendar | *Private* |  |
| toDate | Date till which the report has to be generated for. | Calendar | *Private* |  |
| cardNo | The card number | Long | *Private* |  |
| cardNoList | The list of card numbers | List | *Private* |  |
| rejectedPercent | The percentage of card rejected | Double | *Private* |  |
| approvedPercent | The percentage of card approved | Double | *Private* |  |
| cardRejectedList | List of card rejected/pending | List<CardApplicationTO> | *Private* |  |
| cardApprovedList | List of card approved | List<ApprovedCardTO> | *Private* |  |

|  |  |
| --- | --- |
| **Method** | **public String getCardNumbers()** |
| Purpose and algorithm description | Gets the card numbers. Create a Object of InfyCreditCardWrapper and invoke the **getCardNumbers()** by passing the **fromDate** and **toDate** as parameters.  Display the details in a datatable as a link. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String rejectedReport()** |
| Purpose and algorithm description | Gets the rejected/pending report. Create a Object of InfyCreditCardWrapper and invoke the **rejectedReport()** by passing the **fromDate** and **toDate** as parameters.  Display the details in a datatable. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String acceptedReport()** |
| Purpose and algorithm description | Gets the accepted card application report. Create a Object of InfyCreditCardWrapper and invoke the **acceptedReport()** by passing the **fromDate** and **toDate** as parameters.  Display the details in a datatable. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

|  |  |
| --- | --- |
| **Method** | **public String statusDetails()** |
| Purpose and algorithm description | Gets the percentage of card application approved & card application rejected/pending. Create a Object of InfyCreditCardWrapper and invoke the **statusDetails()** by passing the **fromDate** and **toDate** as parameters.  Display the details as a link. |
| EntityBeans/ ValueObjects Used | InfyCreditCardWrapper |
| Exceptions |  |
| Special Notes | N.A |

# Summary

The Infy Credit Card Inc. (ICCI) is a retail n-tired Java EE based application. The application design is captured in object model and deployment model. UML class diagrams and sequence diagrams are used to document the object model of the application design. Critical things related to design like error handling, exception handling, form validation, presentation logic, business logic and data access logic is captured as a part of the object model.

The deployment model captures the design of infrastructure elements on which ICCI has to be deployed and rolled out. The data design building blocks are captured in the database design document.