Courier Management System

Software Design Document

Version 0.9

*Prepared By*

**Document Identifier**

|  |  |
| --- | --- |
| **Project Code** |  |
| **Project Name** |  |
| **Document Name** |  |
| **Version** |  |
| File Name |  |
| Classification |  |
| Client |  |

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Signature & Date** |
| **Prepared By** |  |  |
| **Reviewed By** |  |  |
| **Approved By** |  |  |
| **Distribution List** |  | |

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Reviewer** | **Change Description** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Overview 5

1.4 Reference Material 5

1.5 Definitions and Acronyms 6

2. System Overview 6

3. System Architecture 7

3.1 Architectural Design 7

4. Data Design 48

4.1 ER Diagram 48

4.2 Data Description 50

5. Component Design 53

6. Human Interface Design 55

6.1 Overview Of User Interface 55

6.2 Screen Images 55

7. Requirement traceability matrix 61

# Introduction

## Purpose

The software design document details how the software requirements should be implemented as well as giving the software development team an overall guidance of the architecture of the software project. The primary stakeholders of the system include registered users, guest users, administrator, pickup staff, delivery staff etc.

## Scope

The project named ‘Courier Management System’ aims at implementing a software system that would manage the essential activities of any courier company and to maintain their details systematically. The traditional manual methods of calculation of rate of consignment, creating periodical reports, billing are made easier, faster and accurate in the proposed system. The system shall allow the user to login from anywhere, track their consignment, complaint about their consignment if any etc. The system also shall enable administrator to generate periodical reports, modify details, and manage complaints. Various functions like pickup, delivery, consignment rate calculation etc. are also included in the system. The ultimate aim is to ensure the smooth and efficient functioning of the courier company by managing its resources and utilities effectively.

## Overview

This document is organized into sections having System Overview, System Architecture, Data Design, Component Design, Human Interface Design, and Requirement Traceability Matrix.

## Reference Material

* IEEE Standard for Software Design Document (IEEE Std 1016-1998)
* CourierManagementSystem\_SRS

## Definitions and Acronyms

**SRS:** Software Requirement Specifications.

**CMS:** Courier Management System.

**Consignment:** The parcel that has to be shipped or delivered.

**Pickup staff:** The user responsible for collecting the consignment from the sender and to ship the consignment.

**Delivery staff:** The user responsible for delivery of consignment.

# System Overview

Courier Management System is an interactive software based system that would almost completely automate the essential processes of courier management system. The various functionalities to be dealt by system are classified into different modules.

The proposed software shall have the following modules or functions:

* + Booking Module
  + Login Module
  + Complaint management Module
  + Report Module
  + Pickup Module
  + Delivery Module
  + Maintenance Module
  + Consignment tracking module

# System Architecture

## Architectural Design

The software architecture of a program or computing system is the structure or structures of the system which comprise the software components, the externally visible properties of those components, the relationships among the components.

Analysis model of CMS comprises of sequence diagrams and collaboration diagrams for all use cases or functionalities identified.

**3.1.1 Sequence diagram and Collaboration diagrams**:

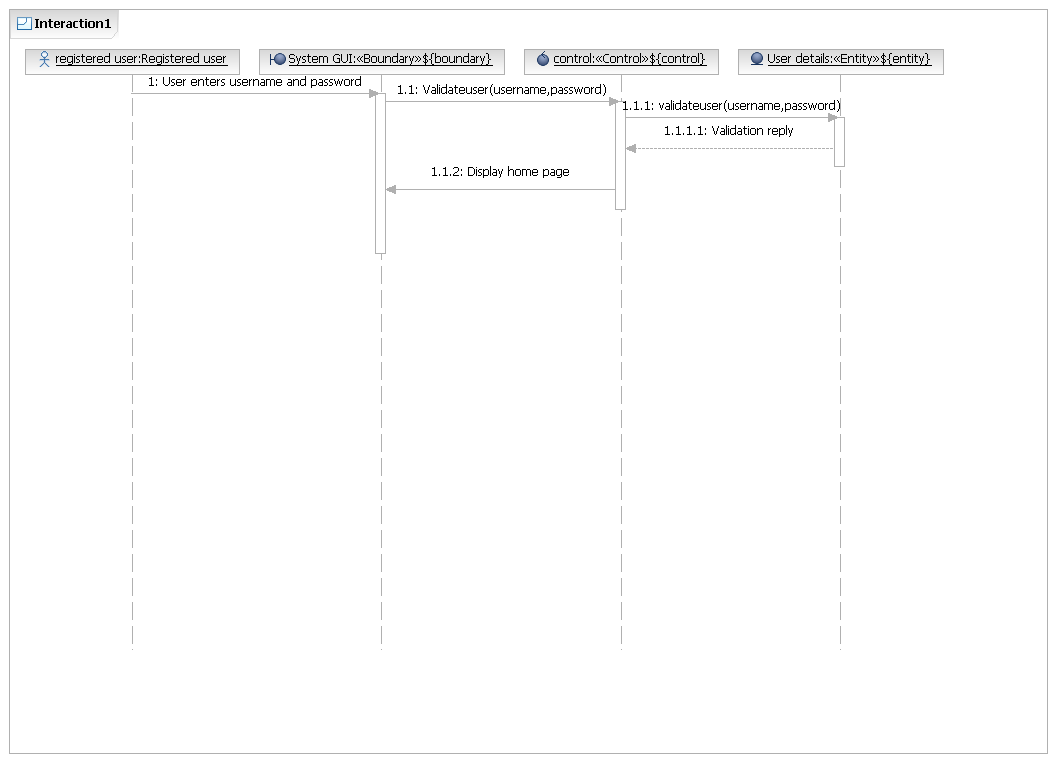
**Sequence diagram**:

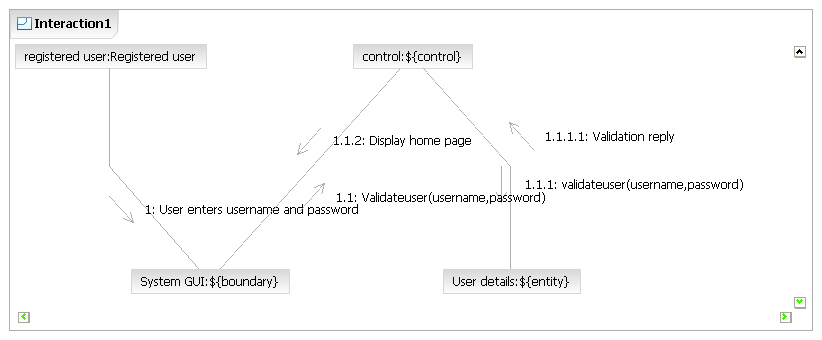
A sequence diagram is a kind of [interaction diagram](http://en.wikipedia.org/wiki/Interaction_diagram) that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**Collaboration diagram**: The UML Collaboration diagram is used to model how objects involved in a scenario interact, with each object instantiating a particular class in the system. Objects are connected by links, each link representing an instance of an association between the respective classes involved. The link shows messages sent between the objects, and the type of message passed.

3.1.1.Login

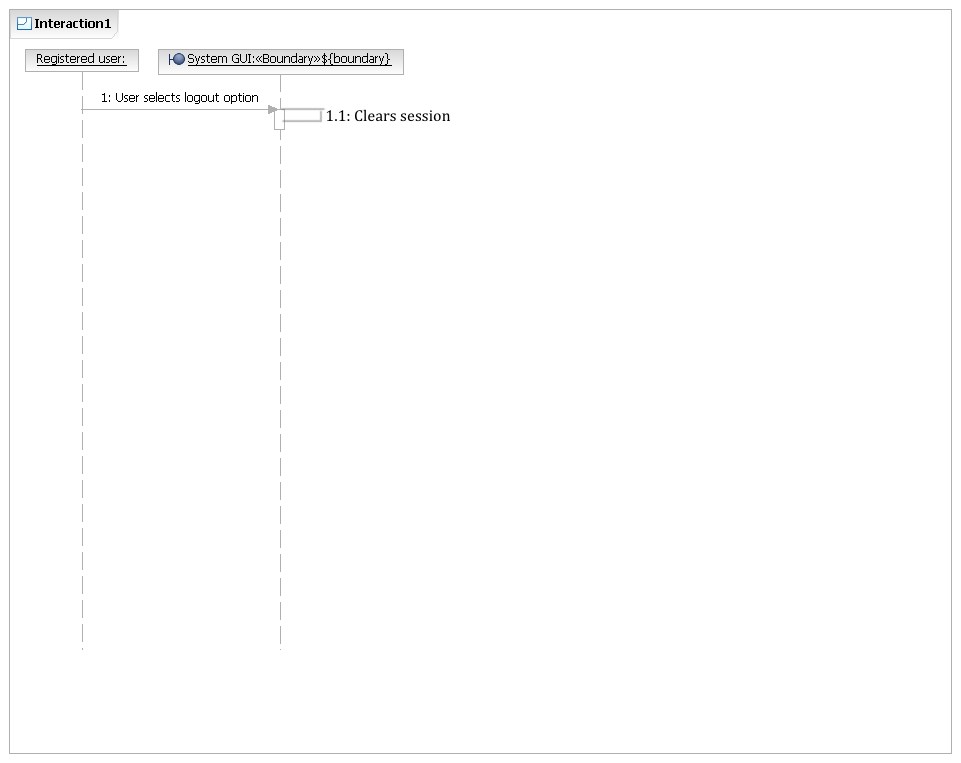
Enables user to login to the system providing user name and password.





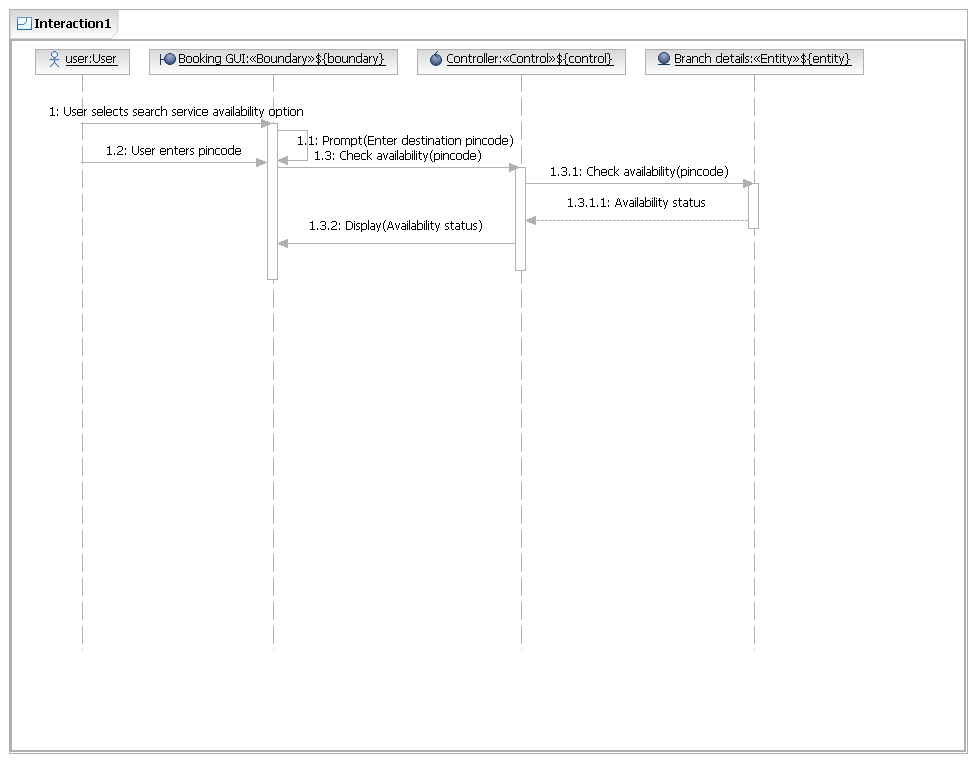
3.1.2.Logout

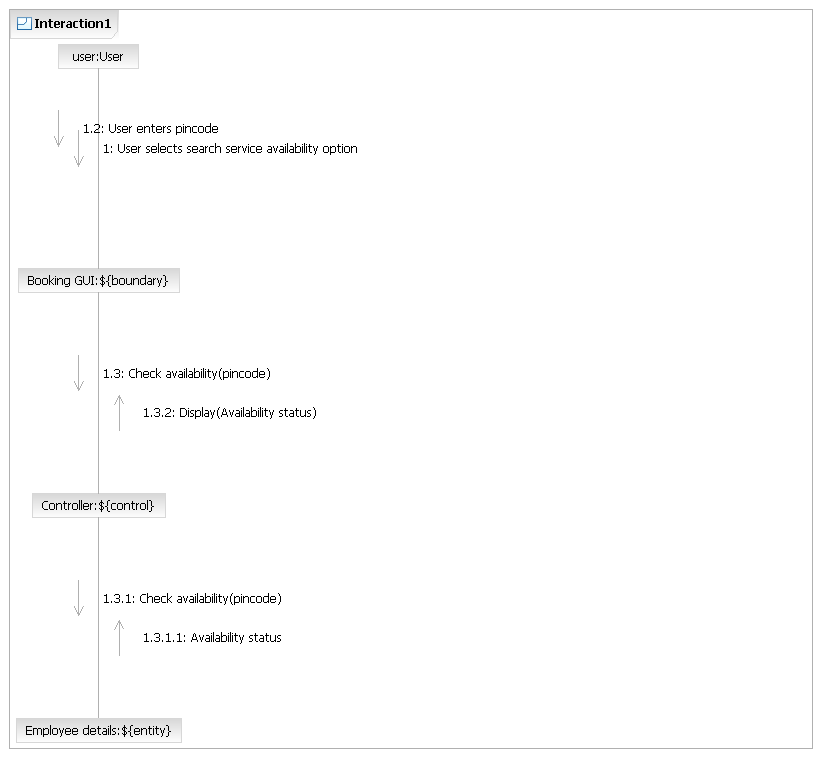
Enables user to logout from the system.



3.1.3.Search service availability

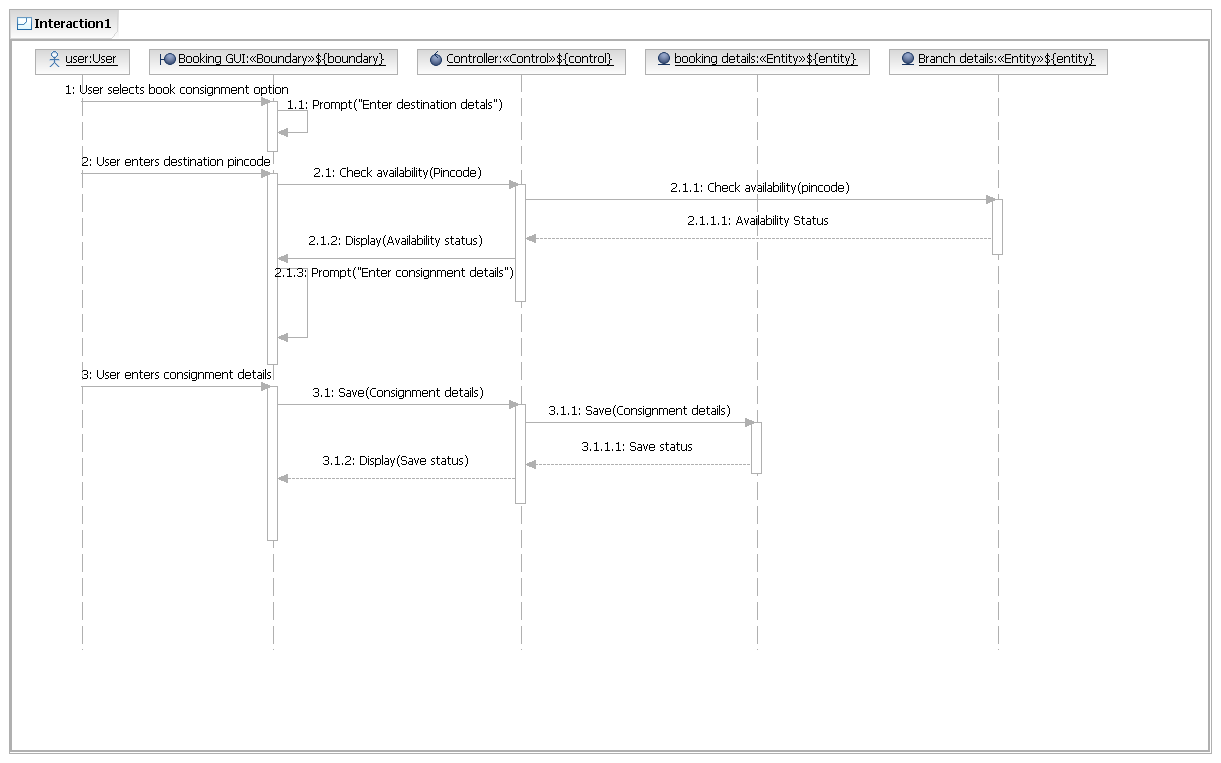
User can search availability of courier service by providing destination pincode.

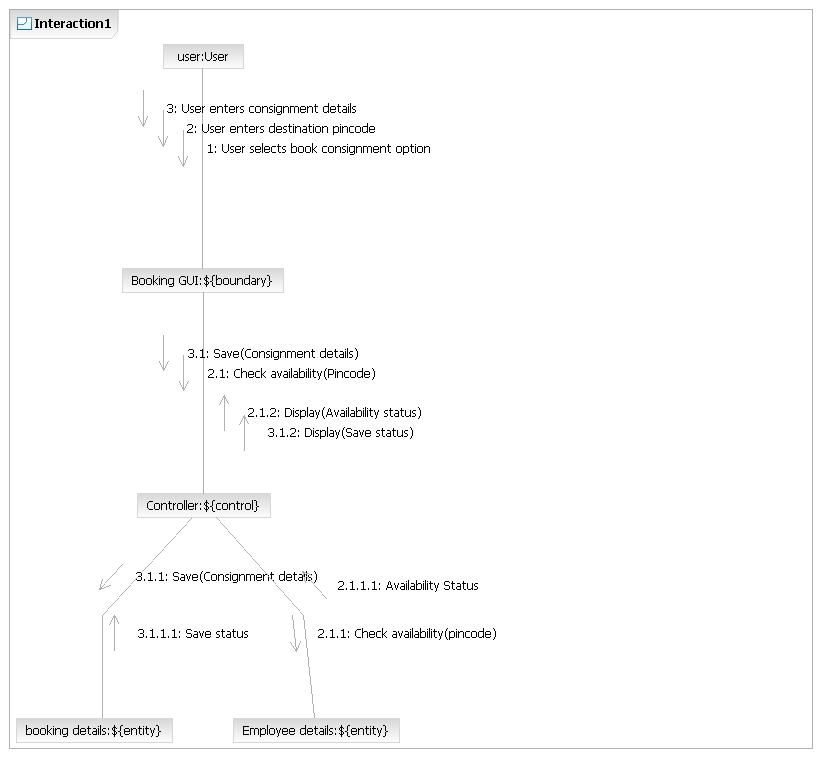




3.1.4.Book consignment

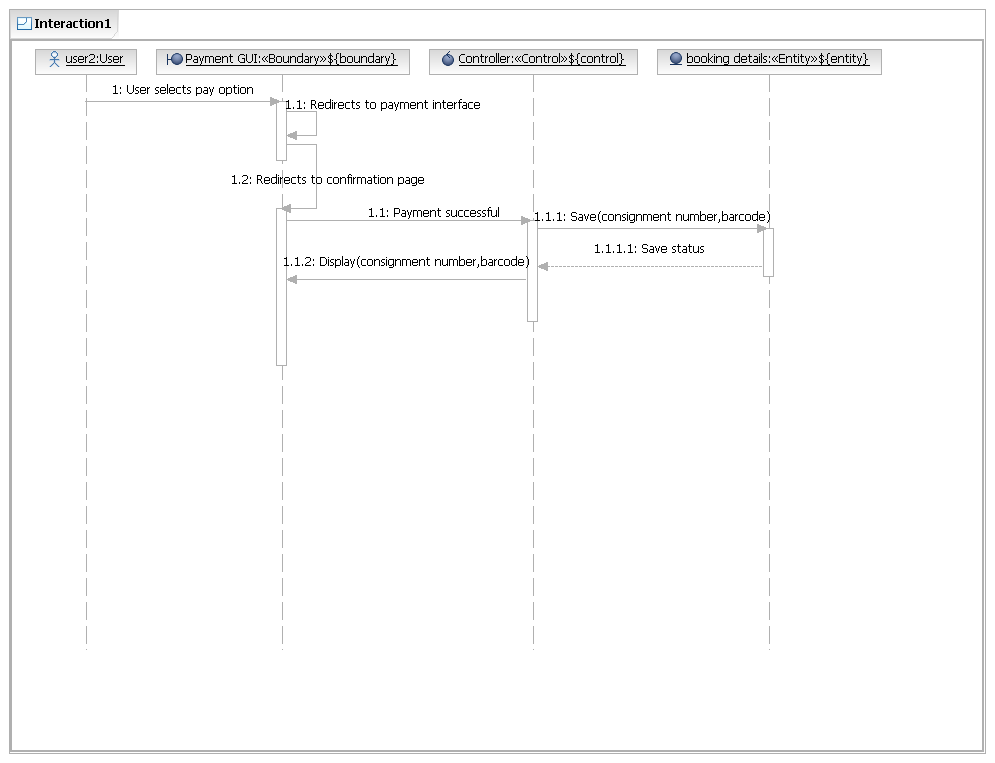
User can book consignment by providing consignment details.

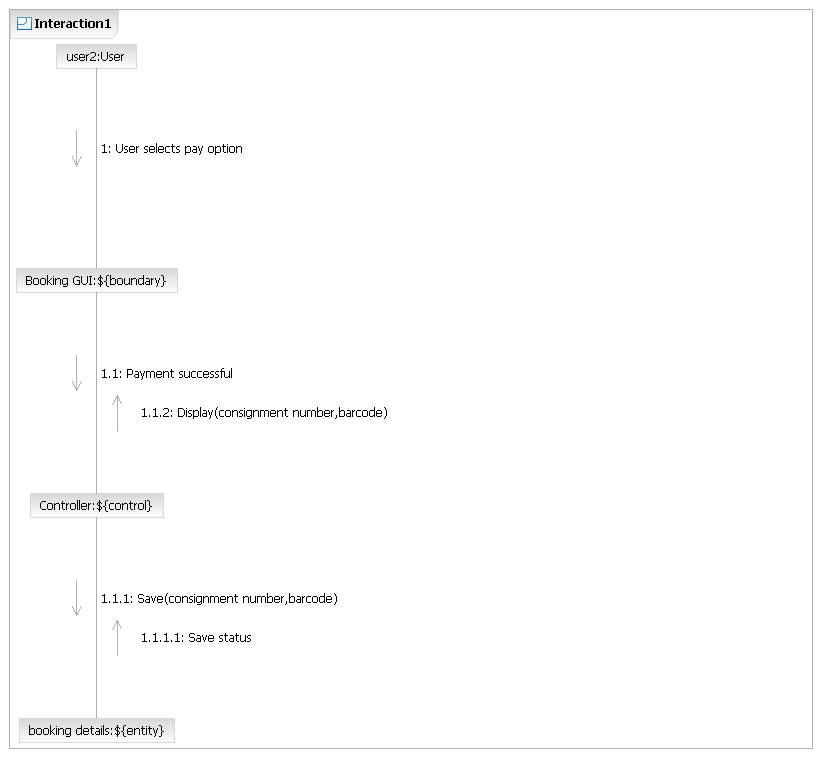




3.1.5.Payment

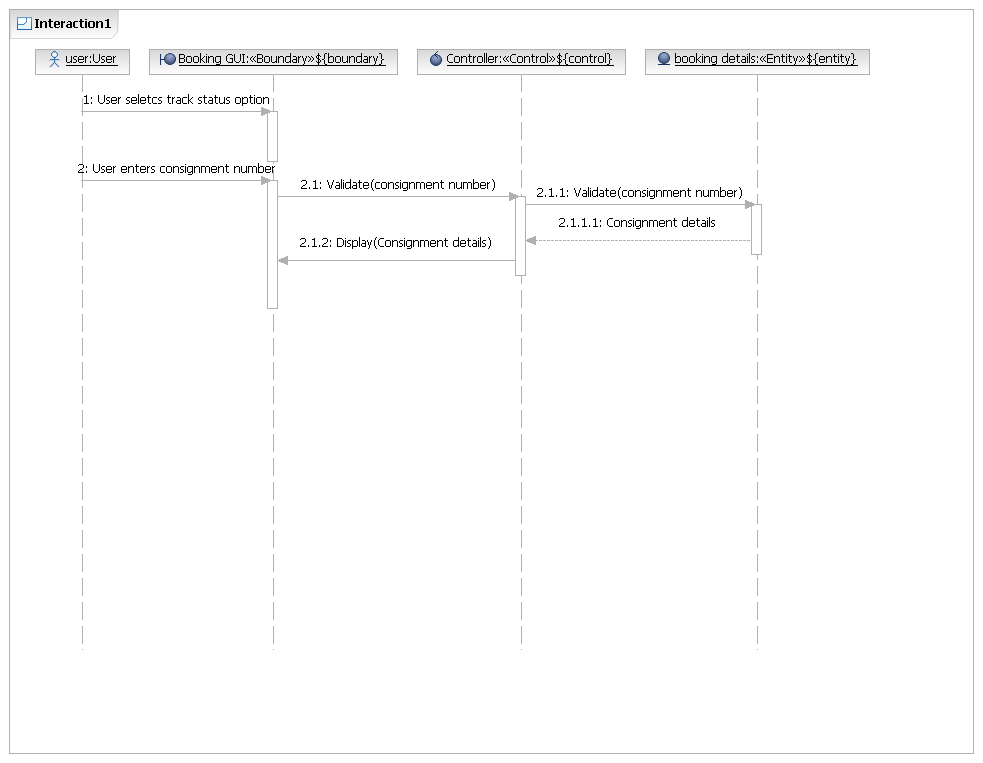
User can make the payment by selecting payment option.

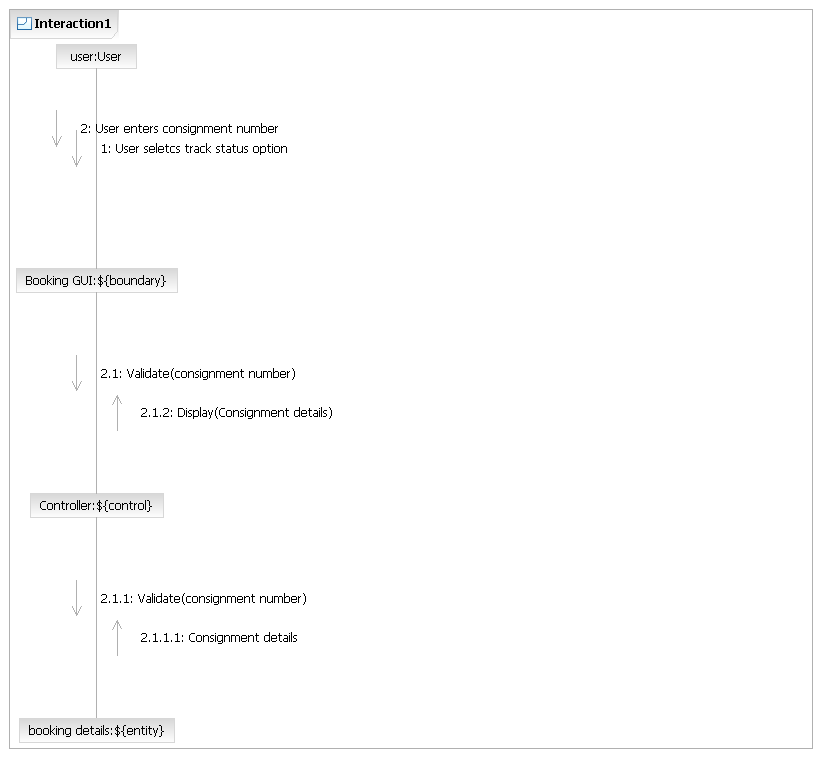




3.1.6.Track status

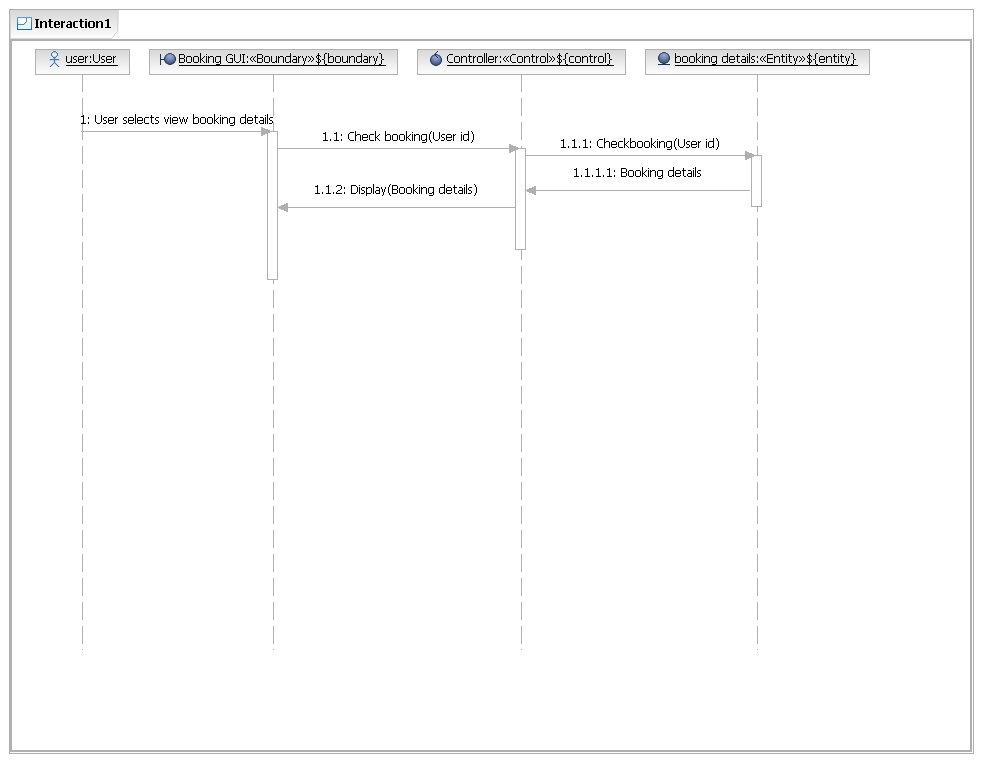
Enables user to track the status of the consignment.

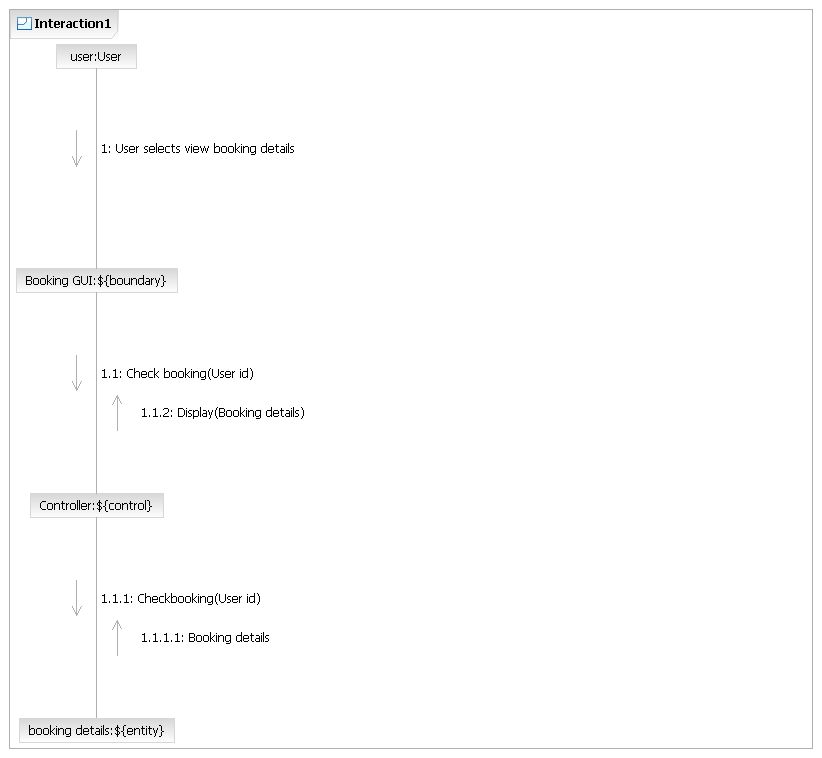




3.1.7.View booking details

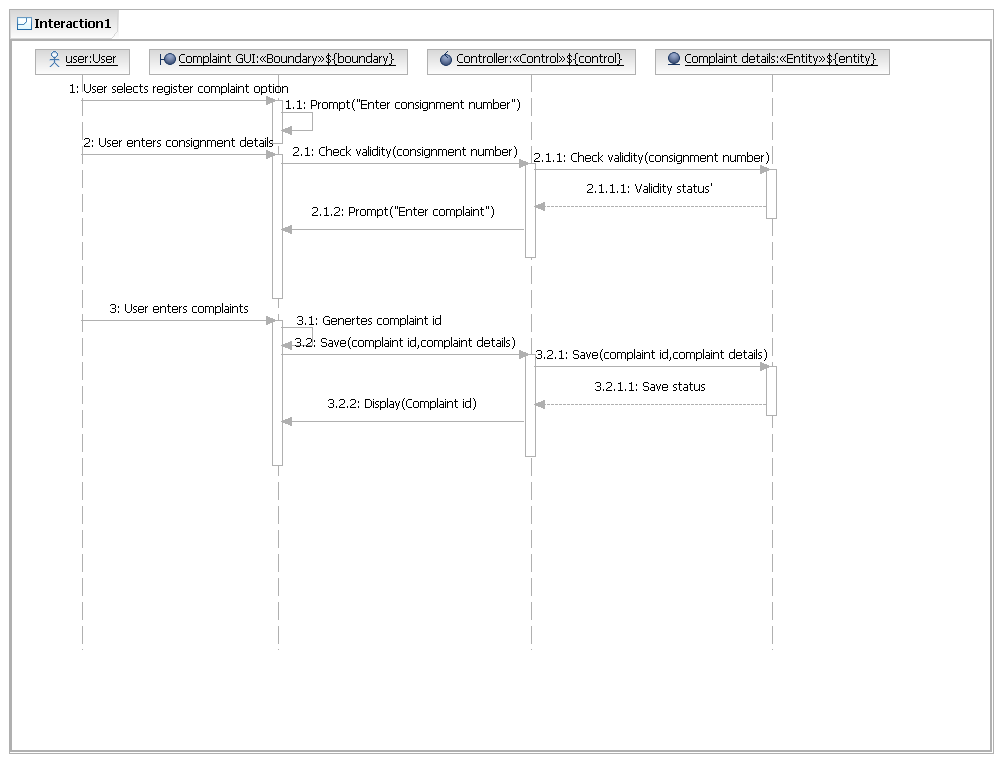
Enables user to view the booking details.

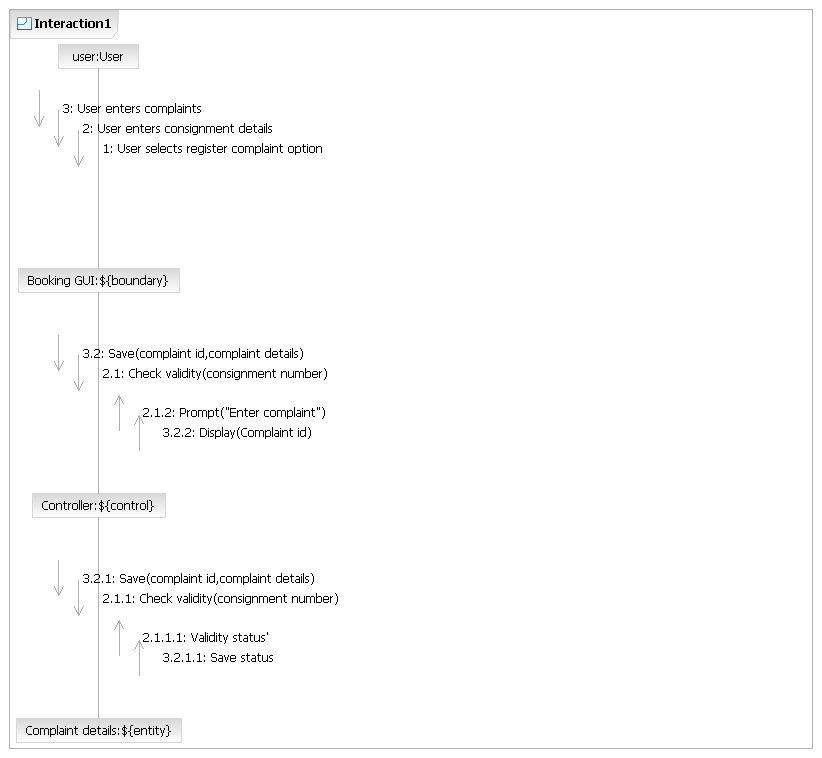




3.1.8.Register complaint

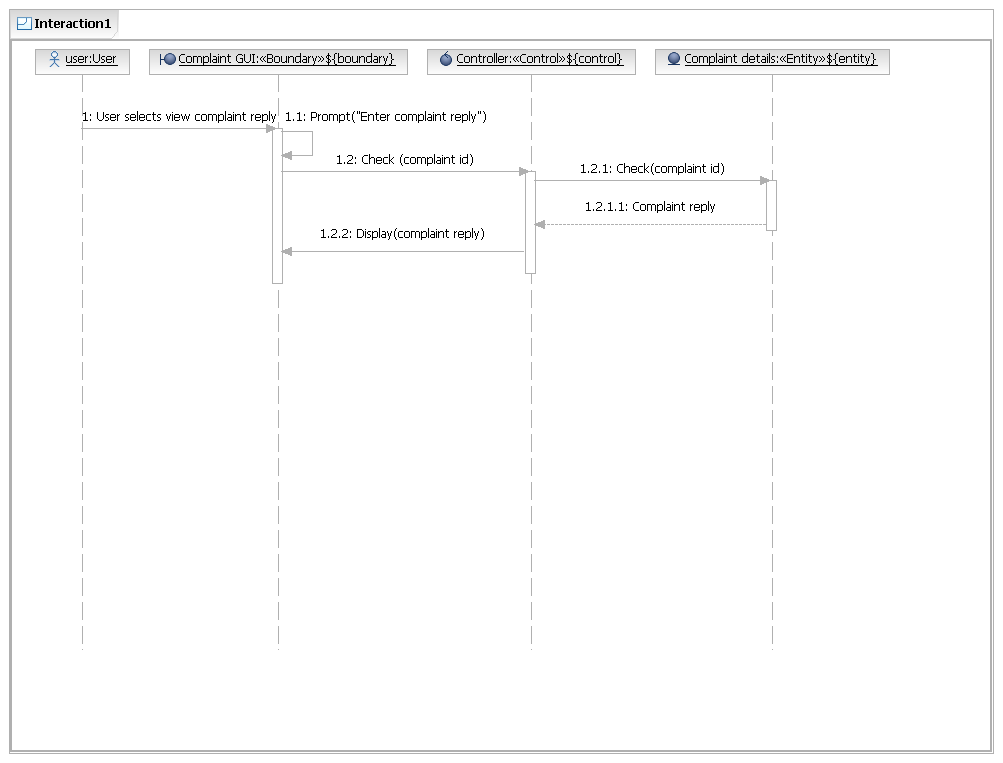
Enables user to register complaint about consignment.

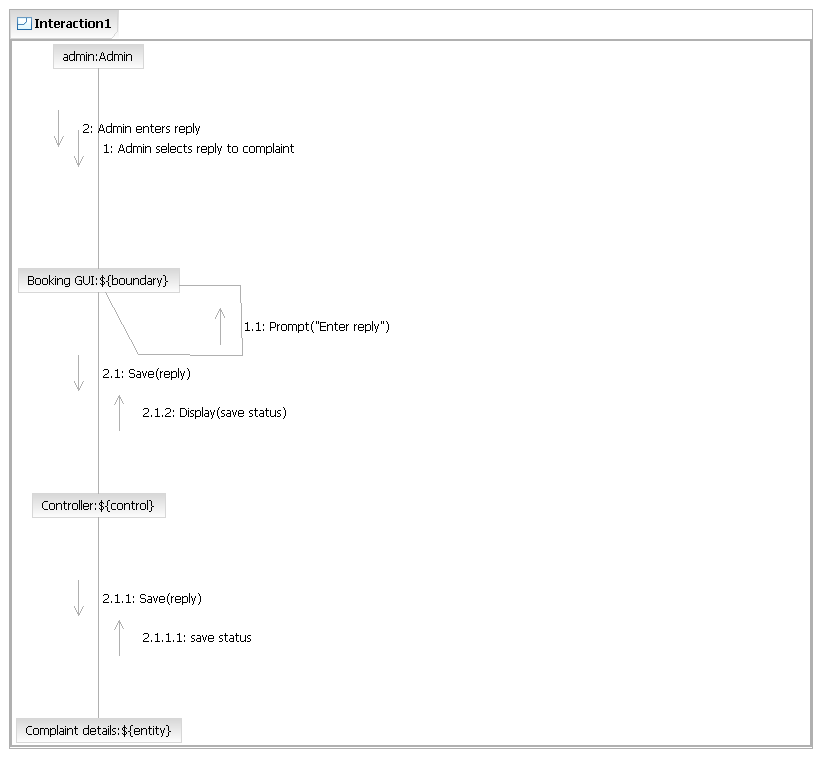




3.1.9.View complaint reply

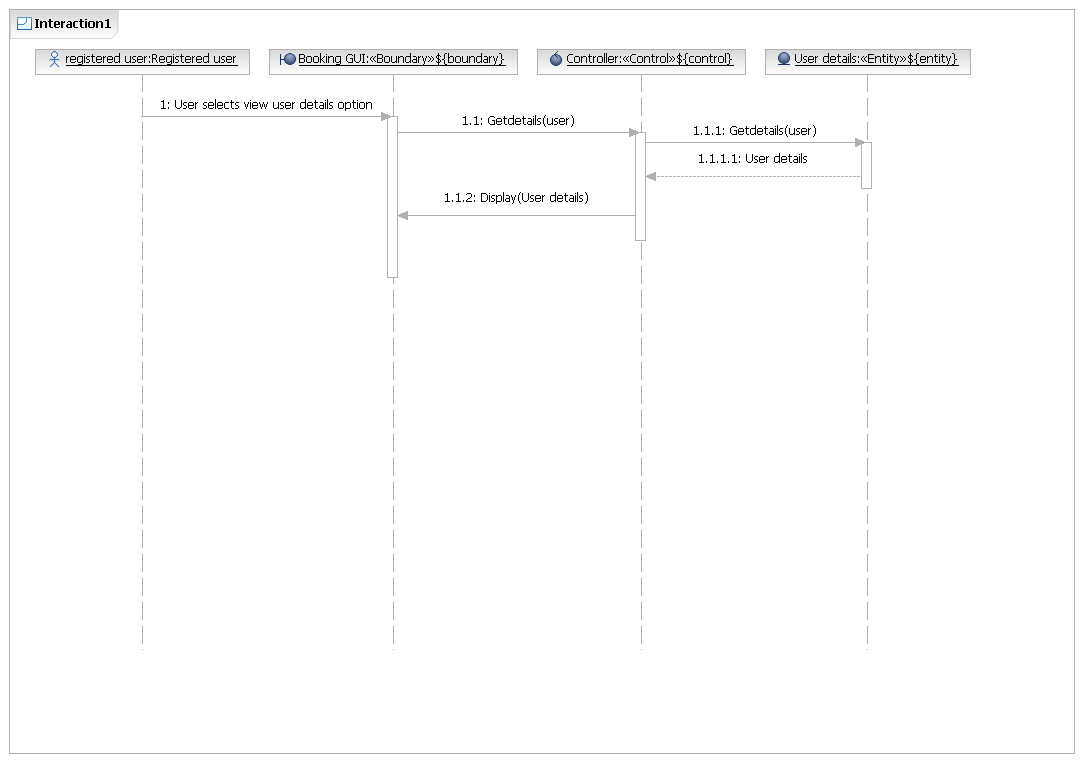
Enables the user to view complaint reply by administrator.

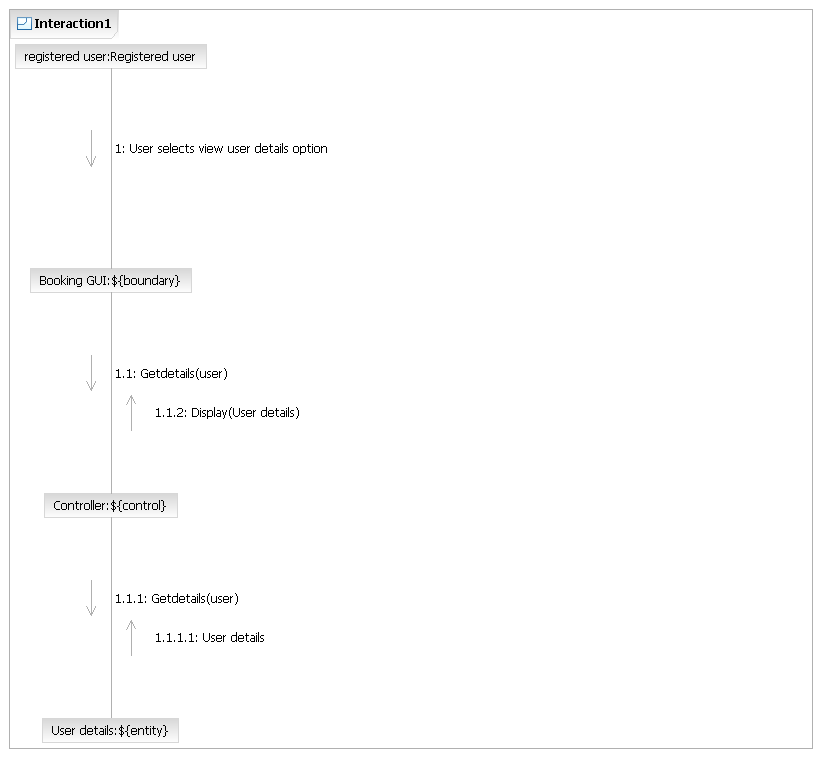




3.1.10.View user details

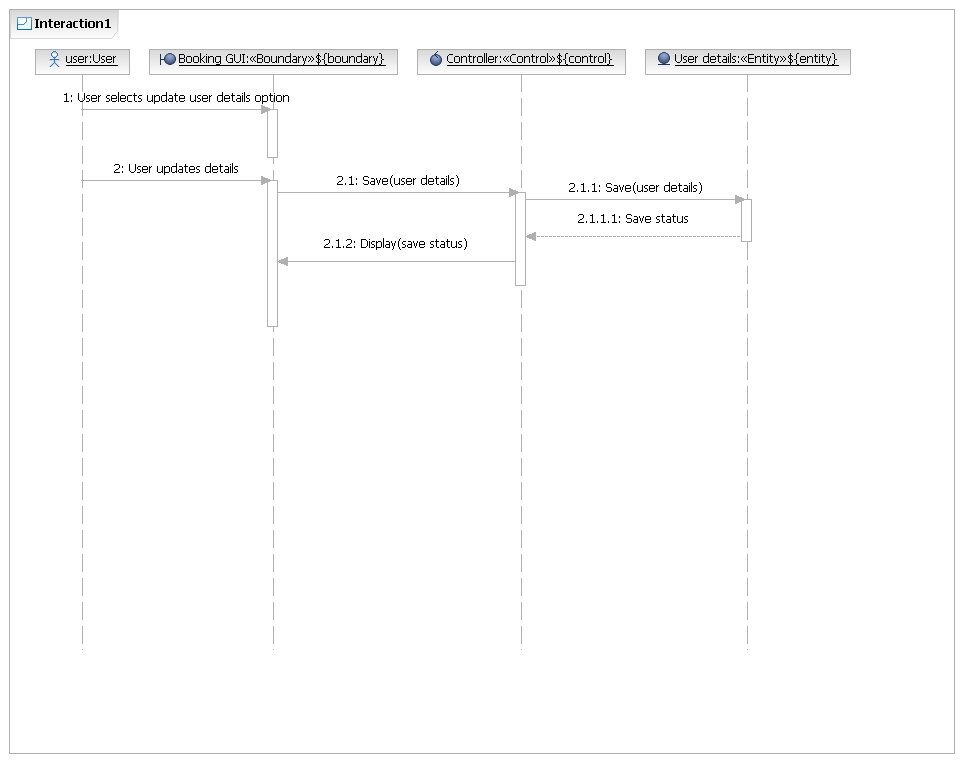
Enables user to view the personal details.

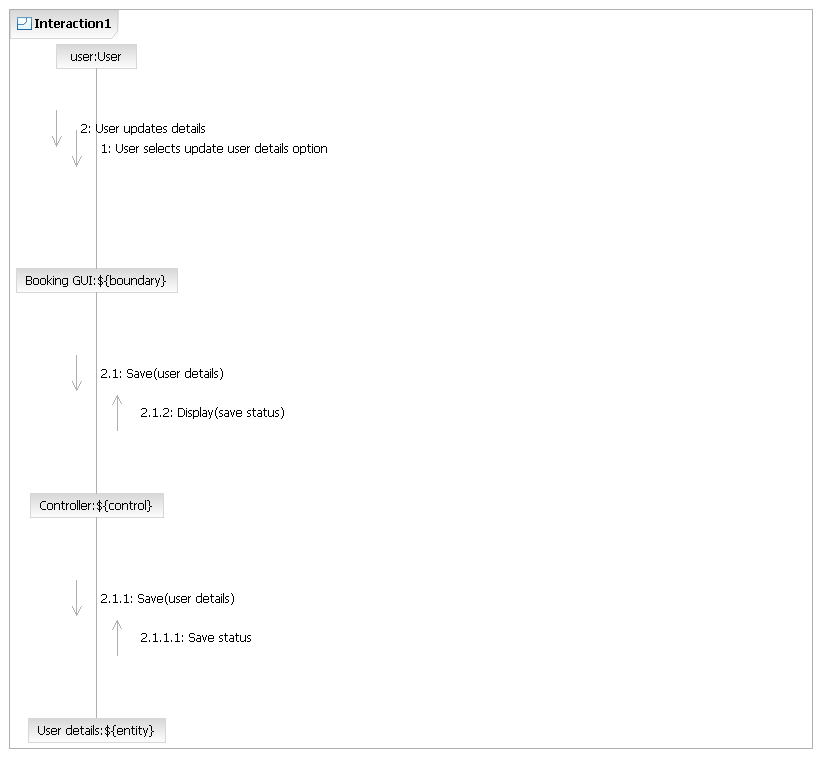




3.1.11.Update user details

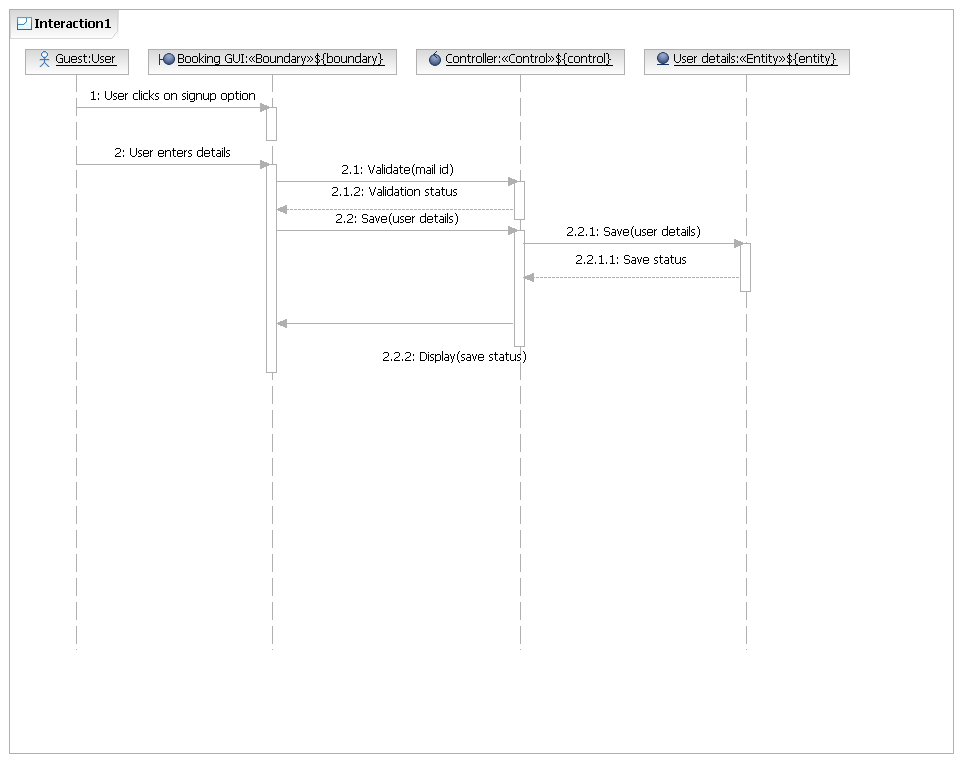
Enables user to update the personal details

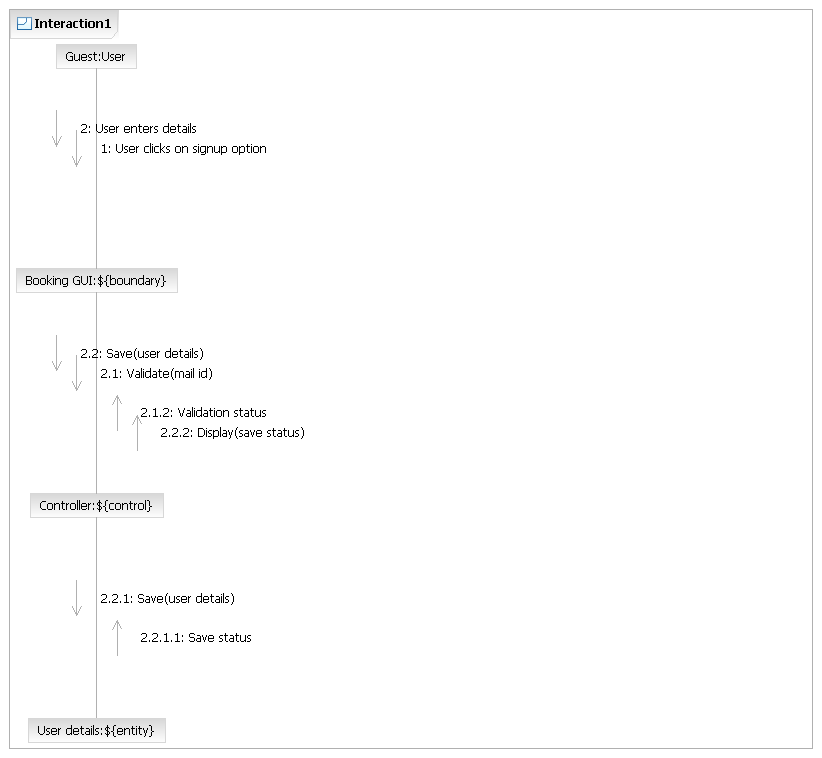




3.1.12.Signup

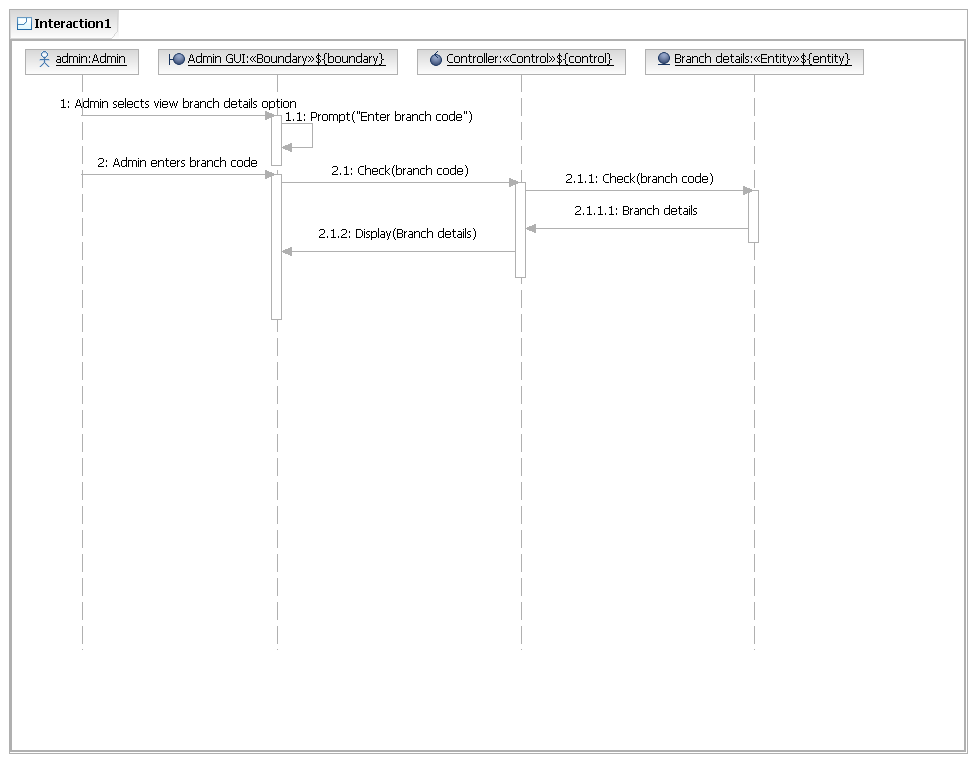
Enables user to signup to the system by providing user details and a valid mail id.

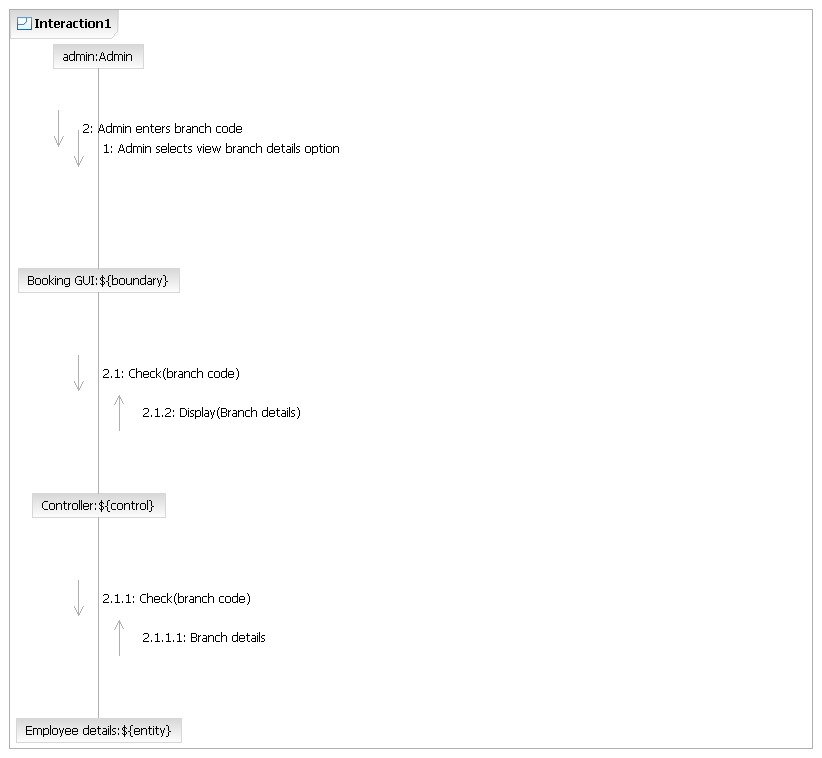




3.1.13.View branch details

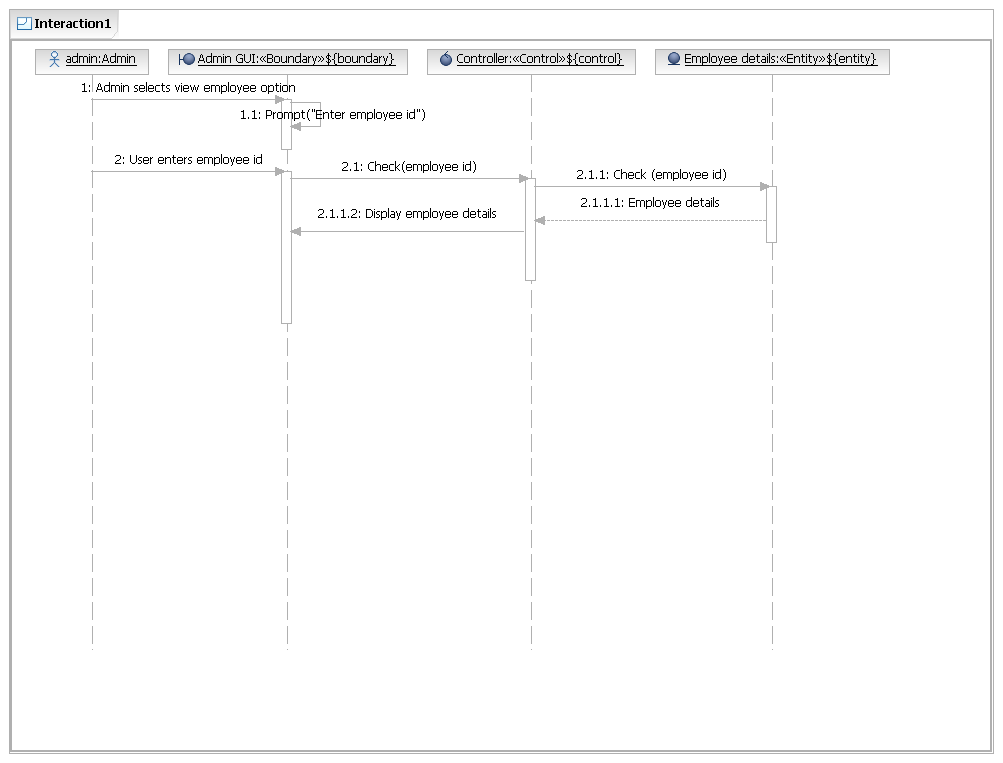
Enables user to view branch details.

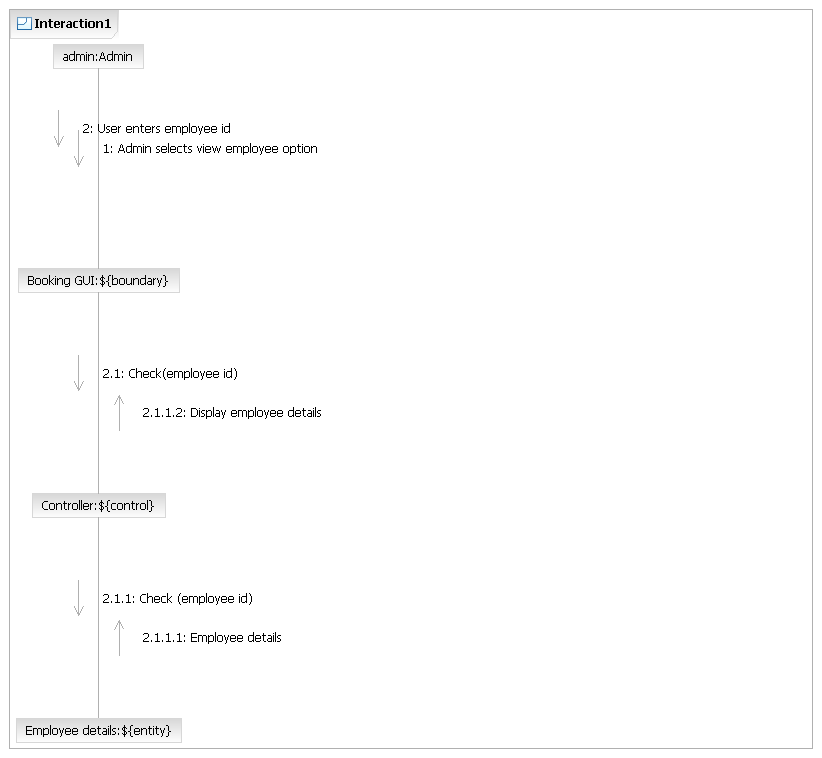




3.1.14.View employee details

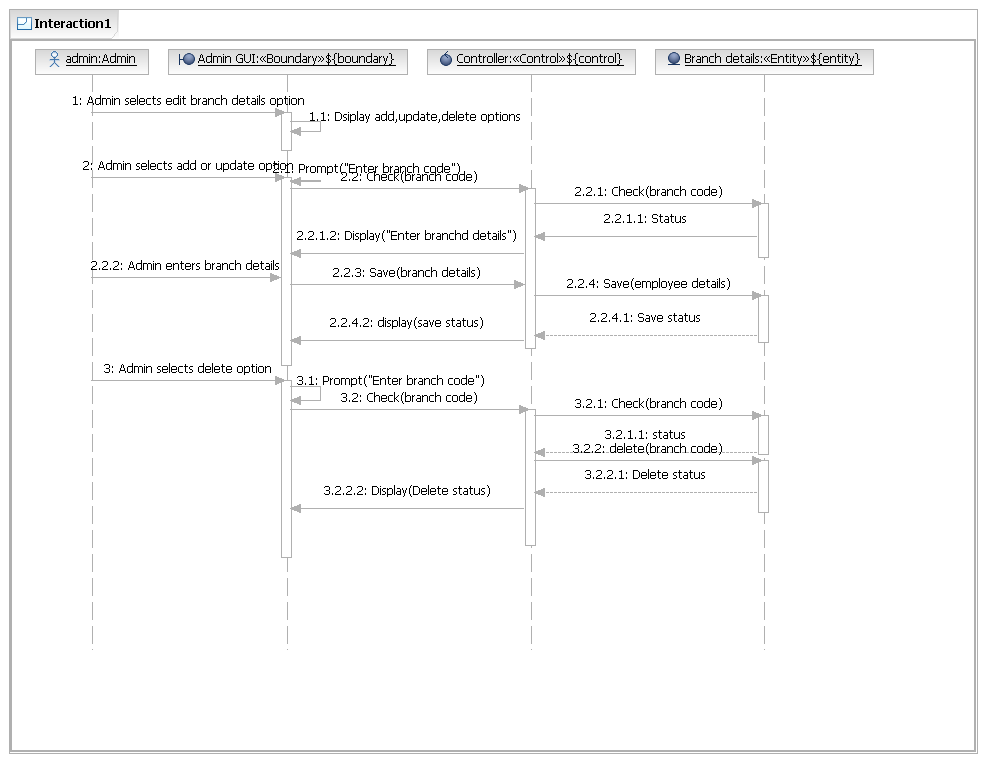
Enables user to view employee details.

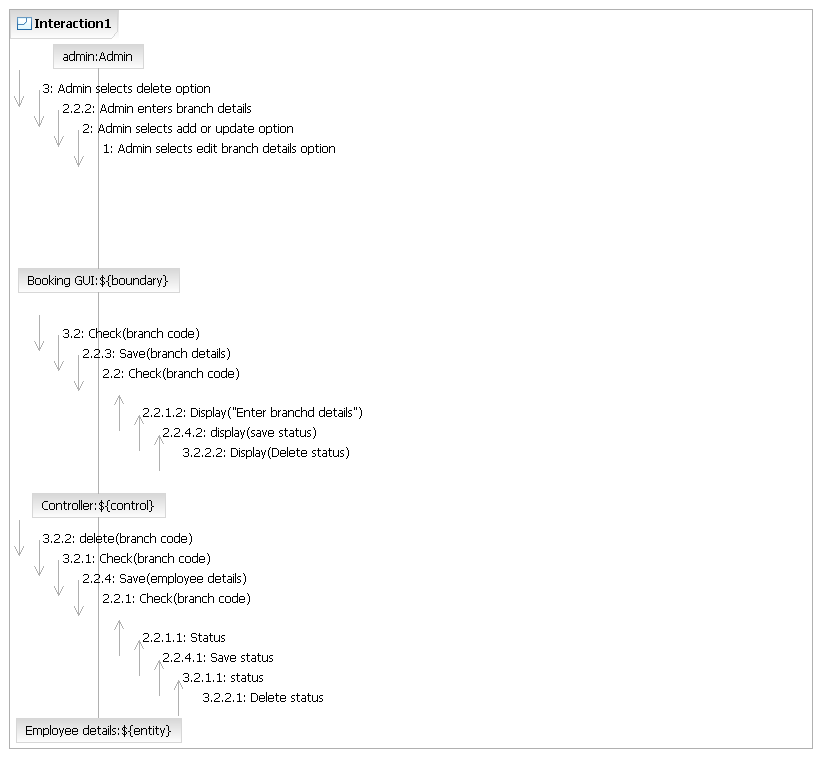


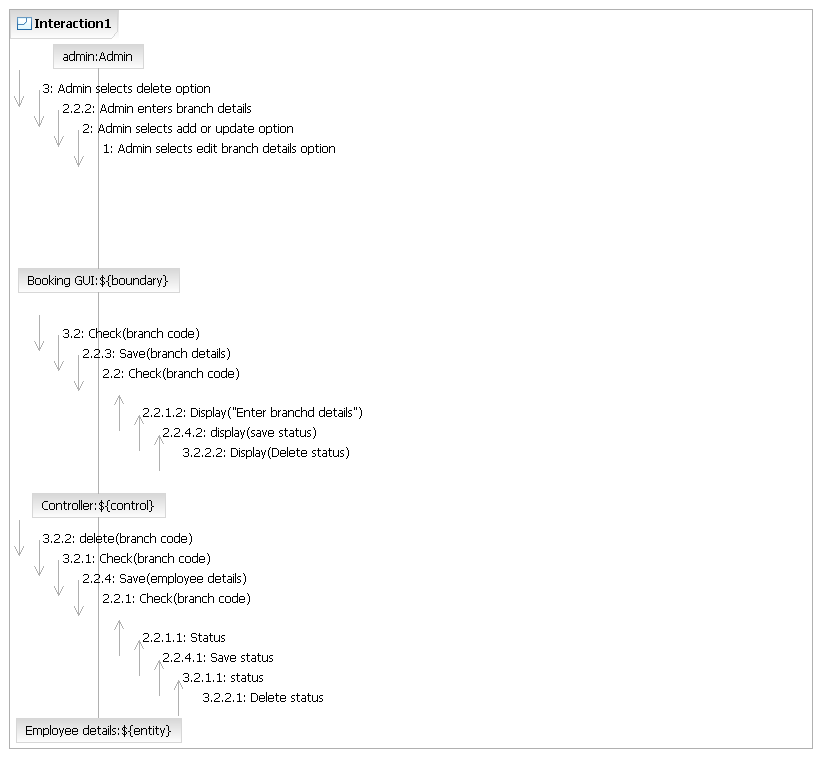


3.1.15.Edit branch details

Enables user to add, update or delete branch details.

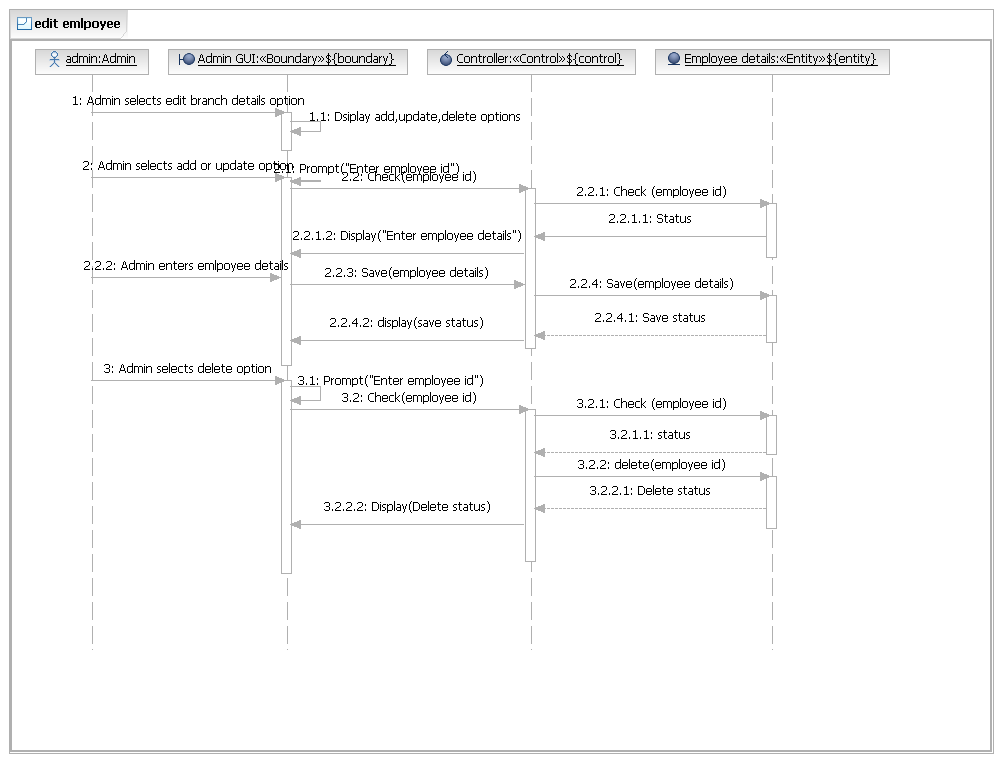


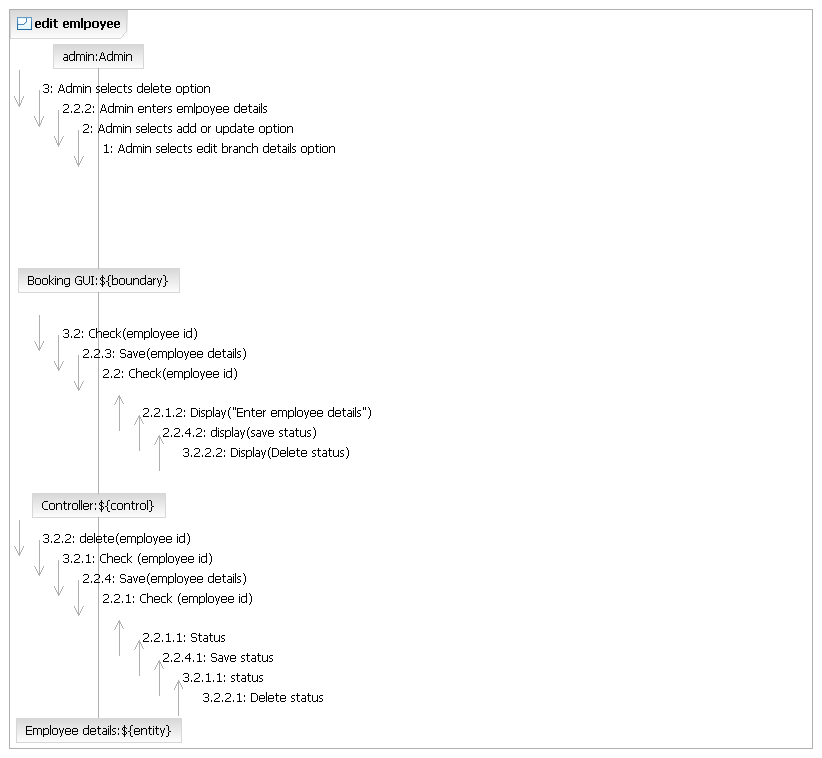




3.1.16.Edit employee details

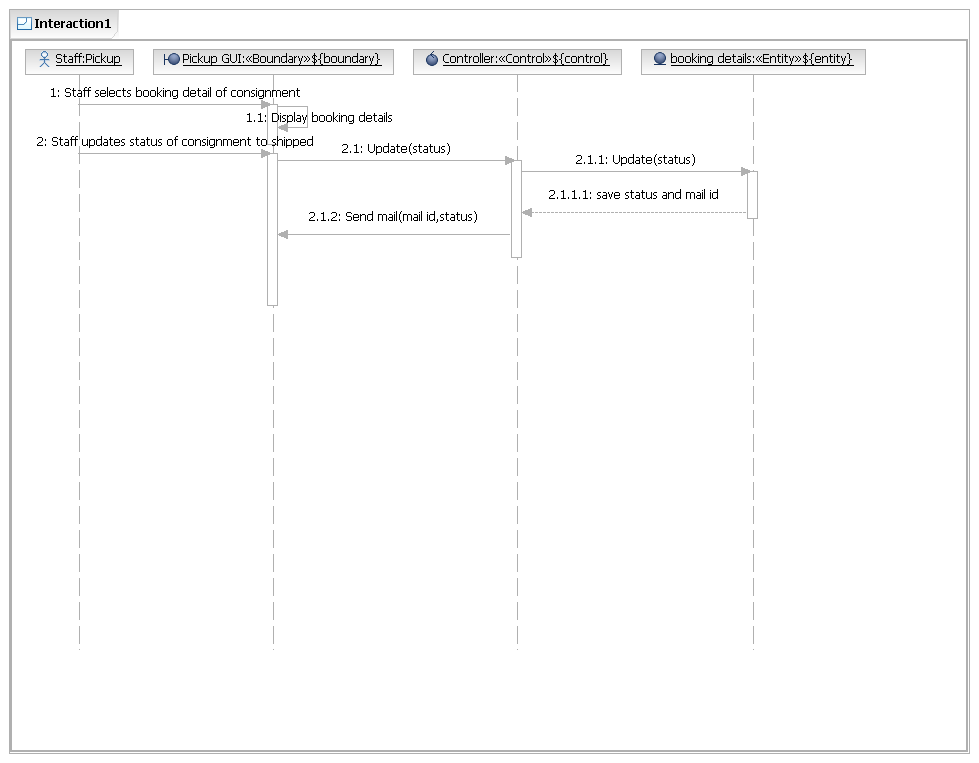
Enables user to add, update or delete employee details.

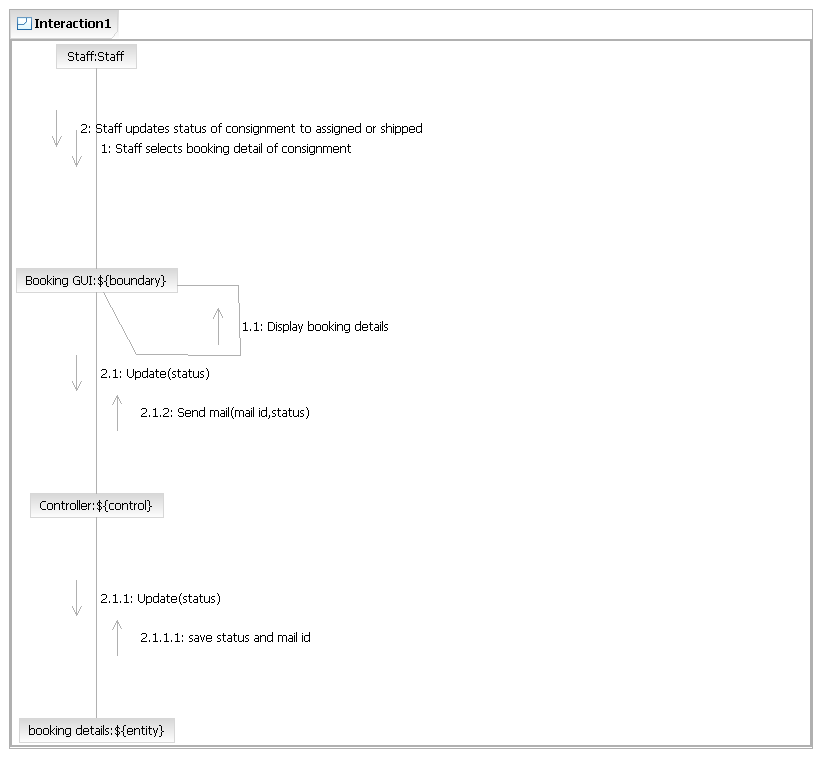




3.1.17.Pickup

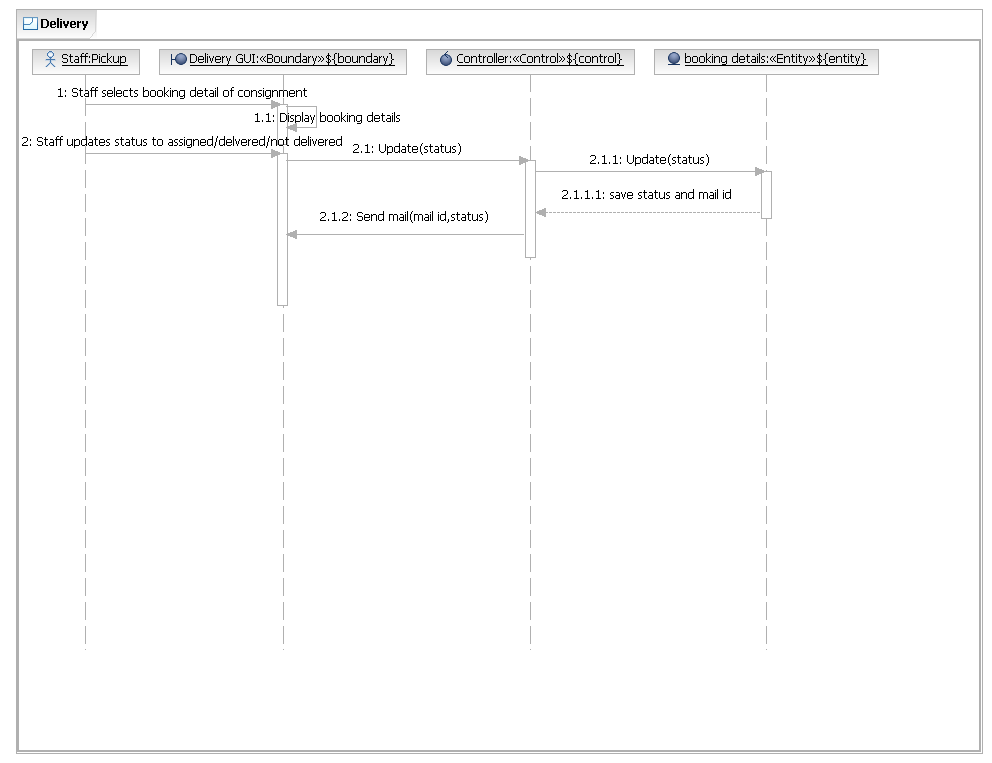
Staff picks up consignment and updates the consignment status.

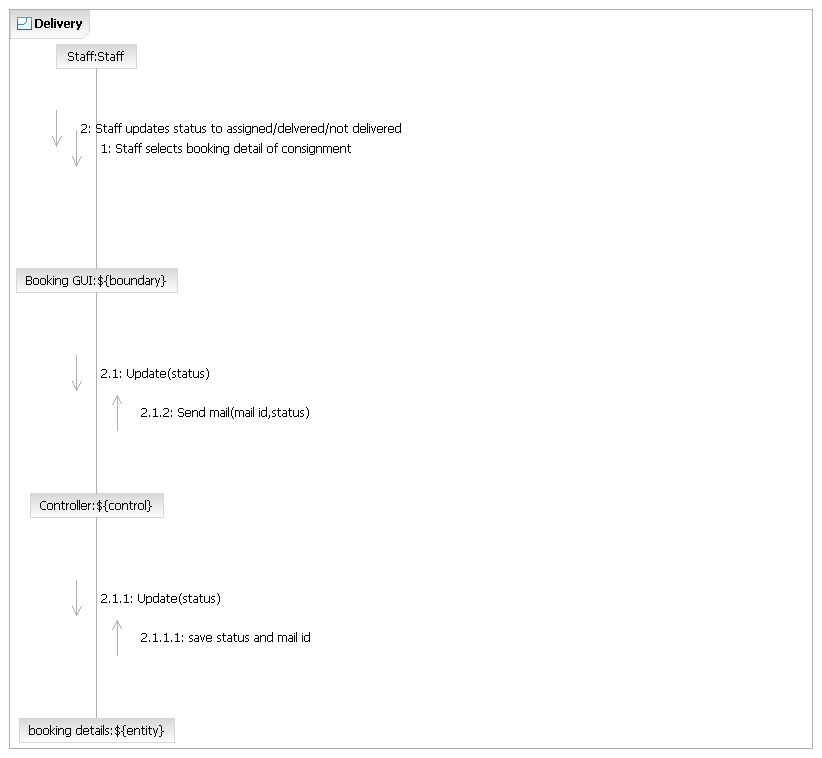




3.1.18.Delivery

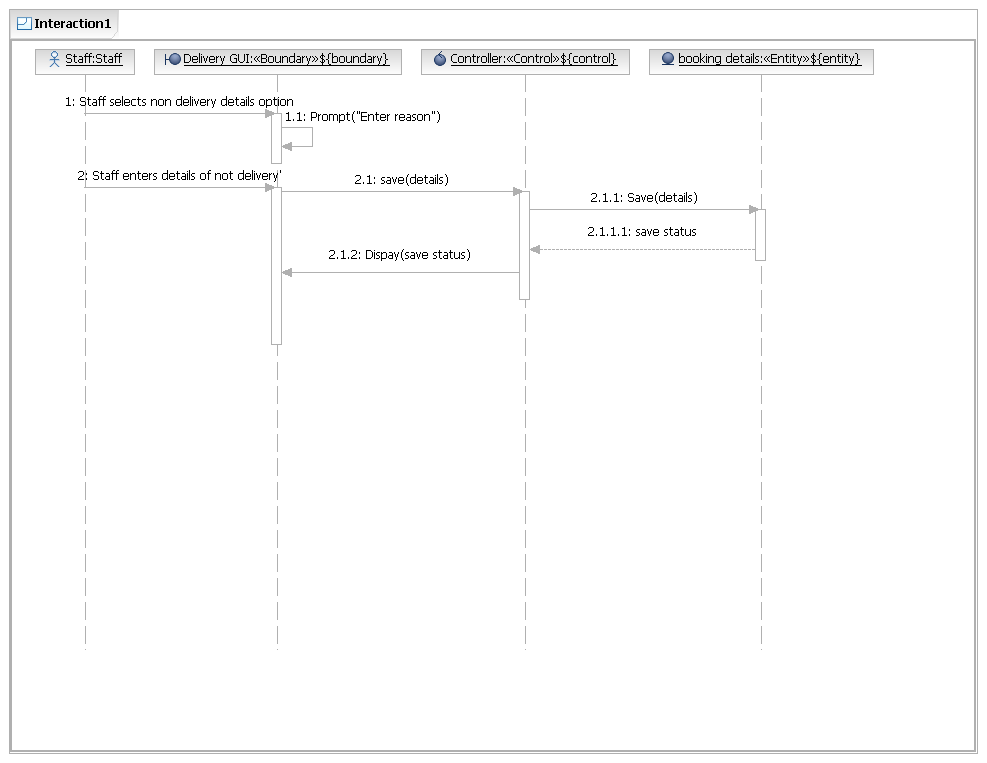
Staff delivers consignment and updates the consignment status.

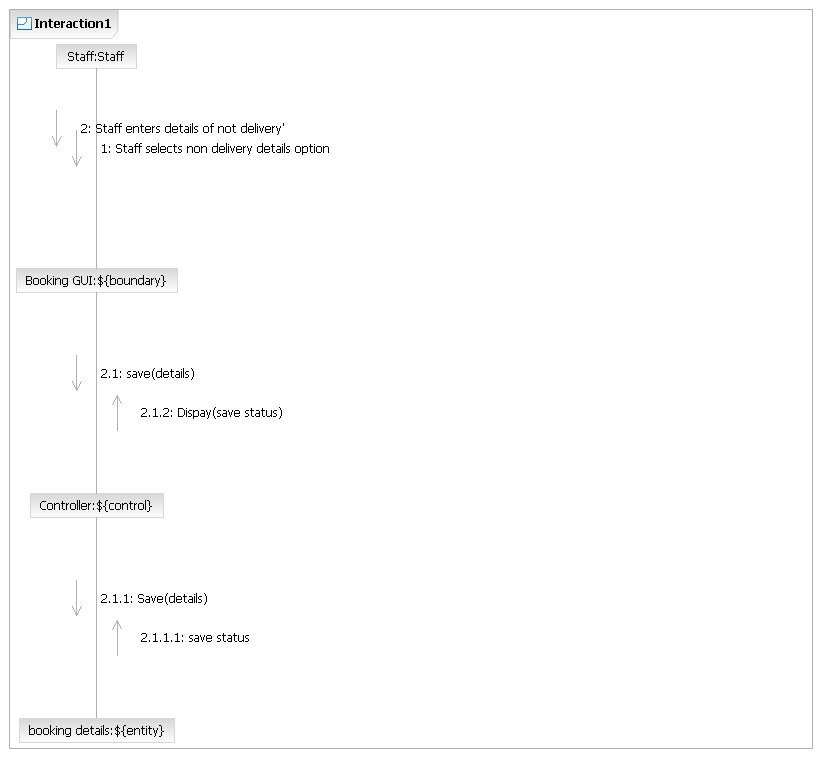




3.1.19.Non delivery details

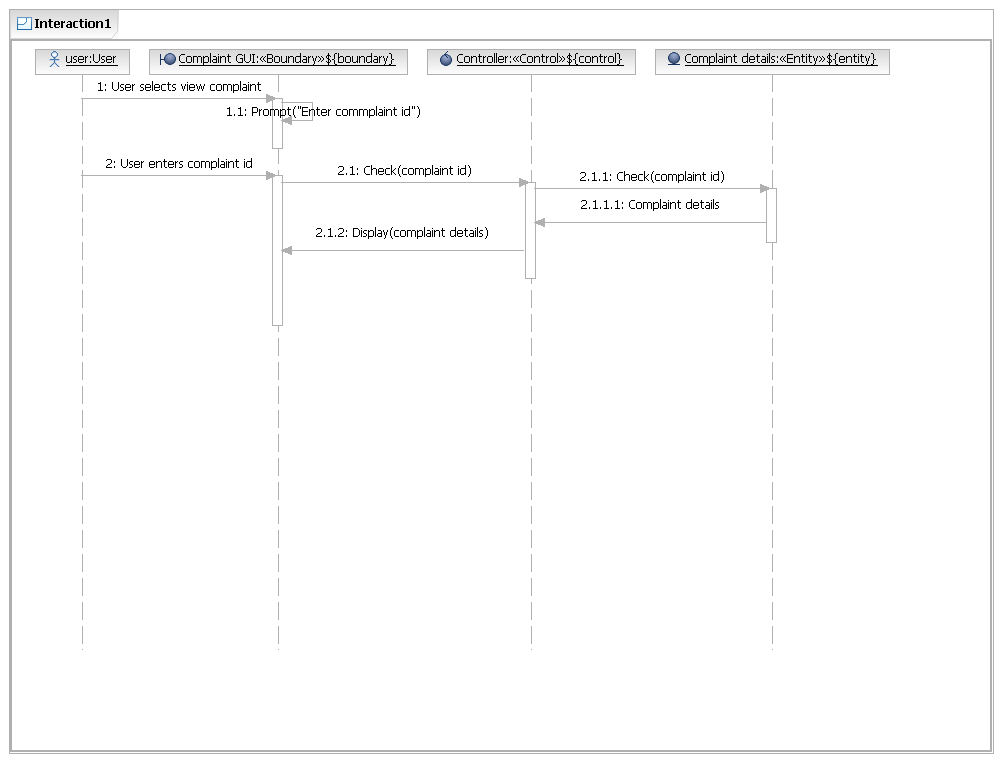
Staff updates the consignment non delivery details.

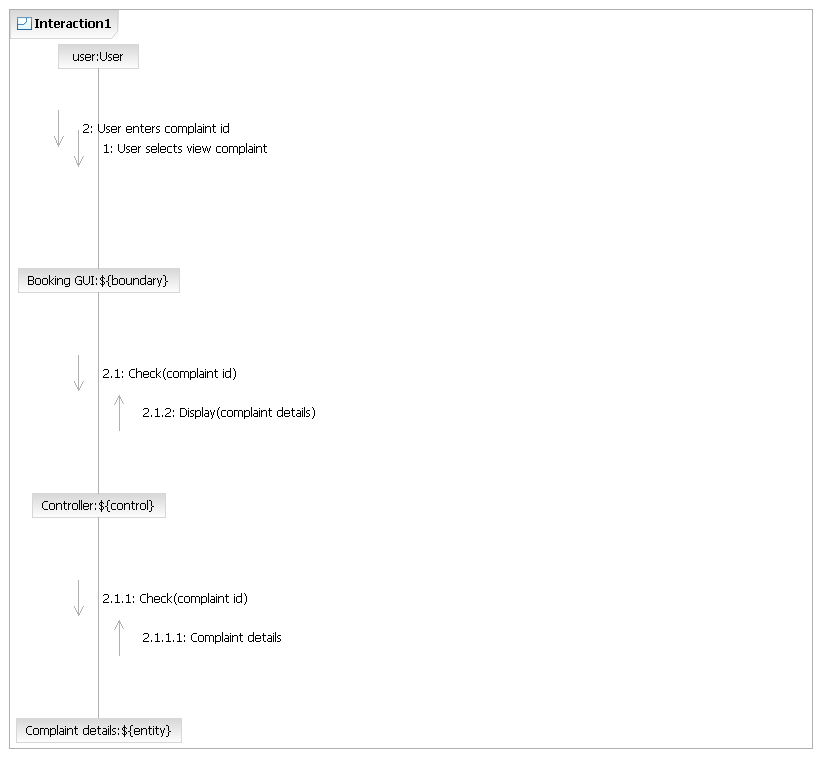


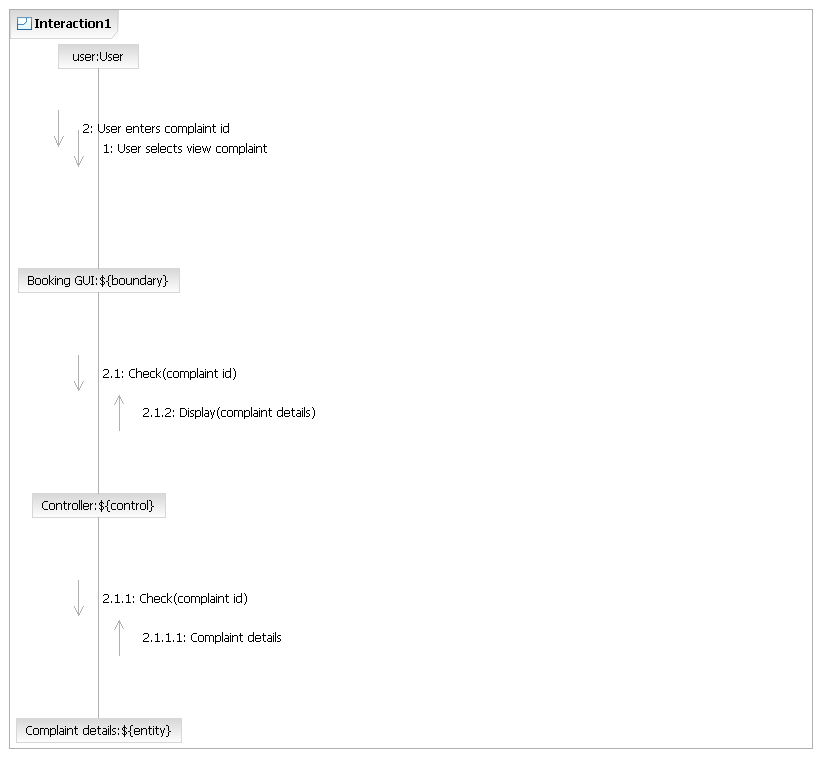


3.1.20.View complaint

User can view the complaint.

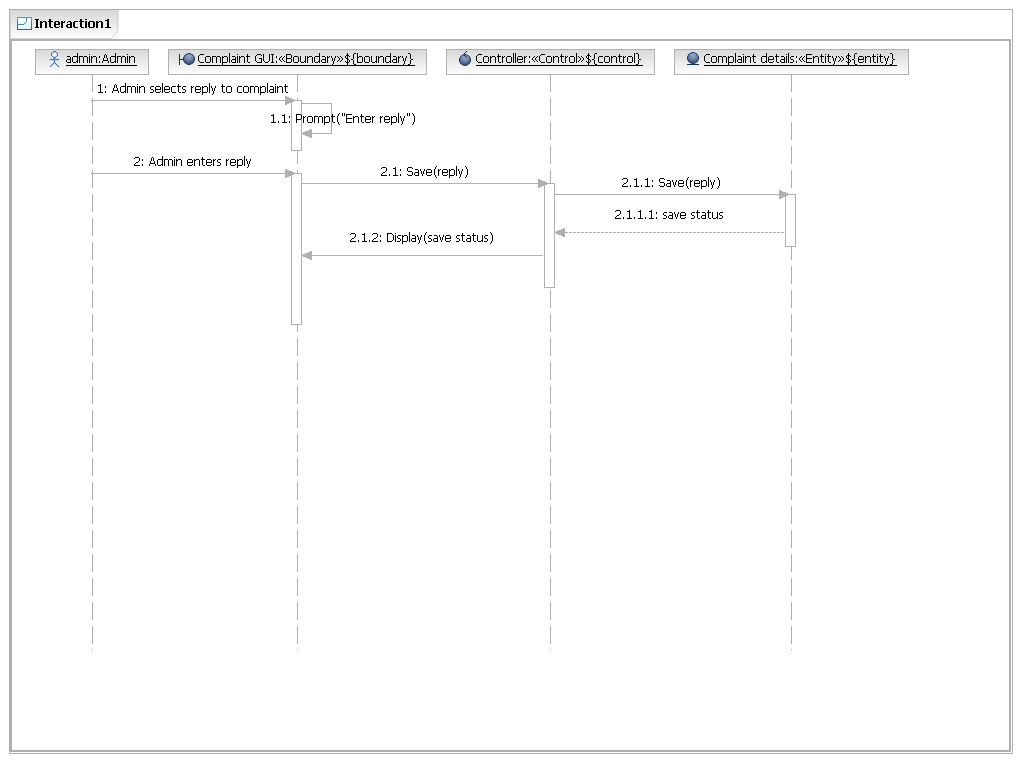


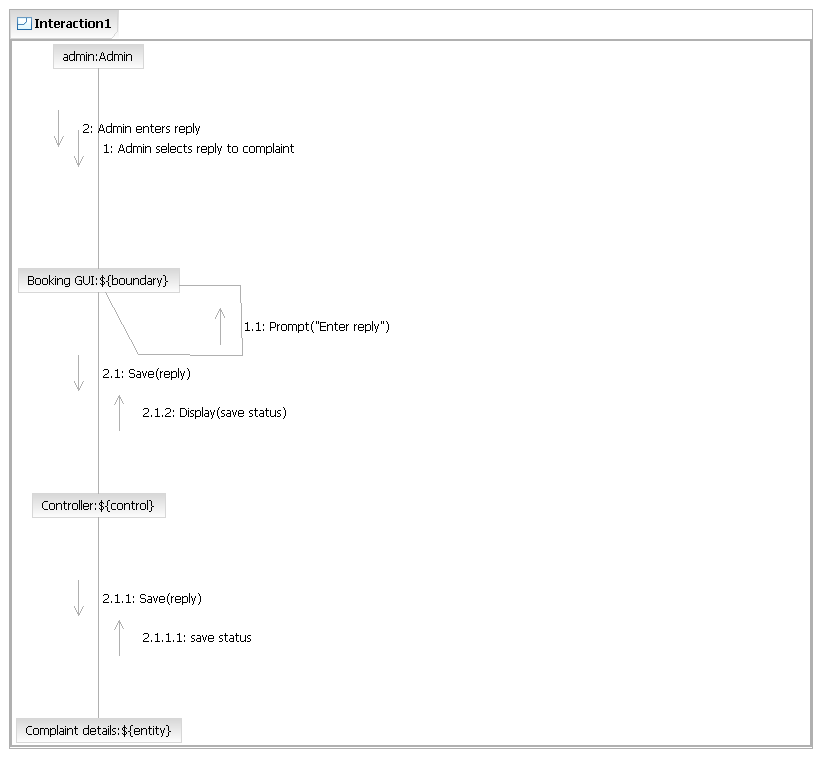




3.1.21.Complaint reply

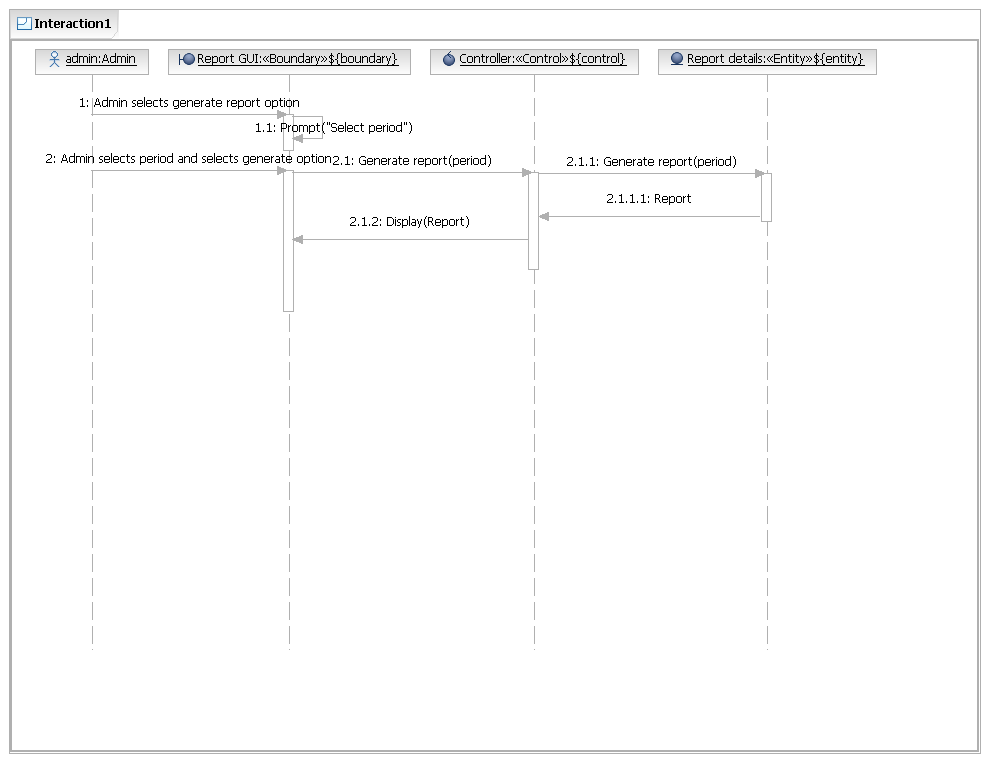
Enables administrator to reply to the complaints.

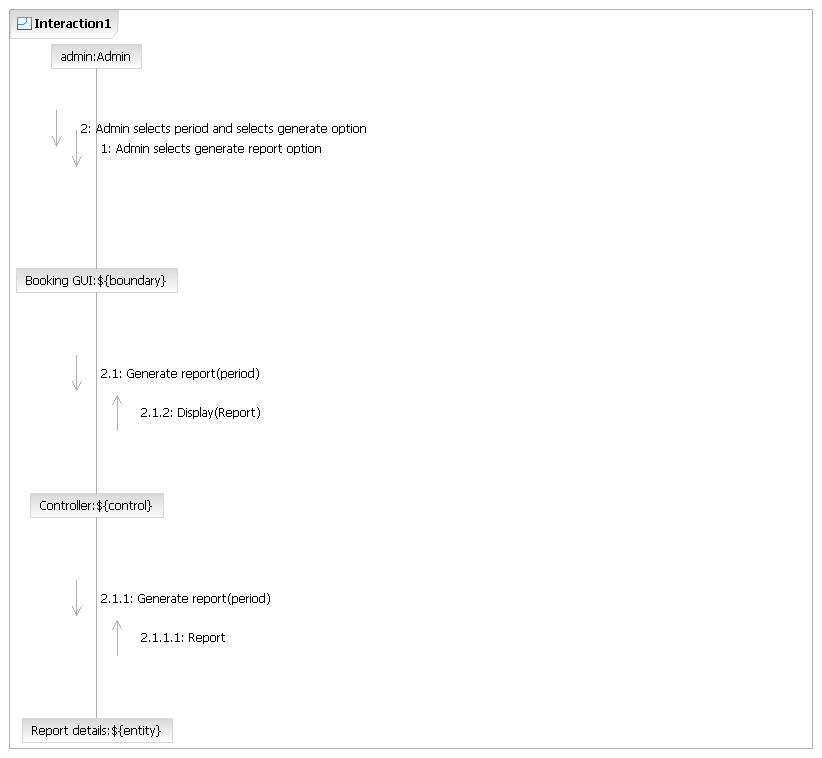




3.1.22.Generate report

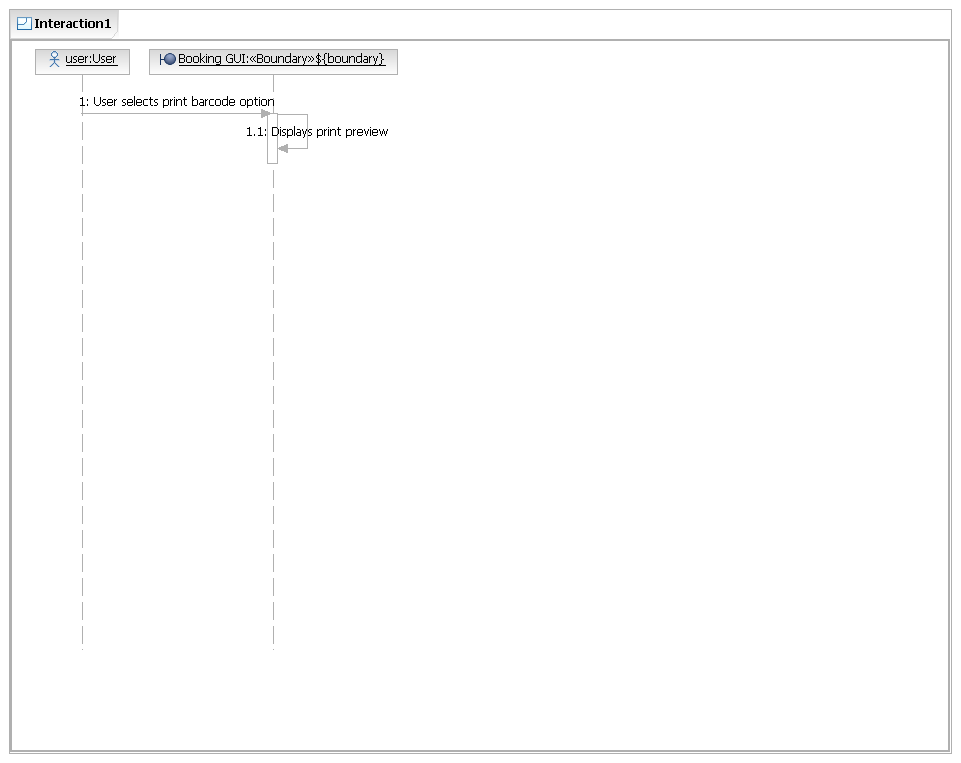
Enables user to generate report for a particular period.

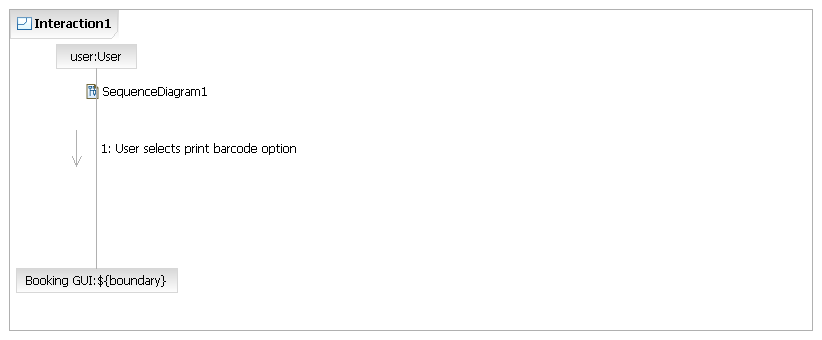




3.1.23.Print barcode

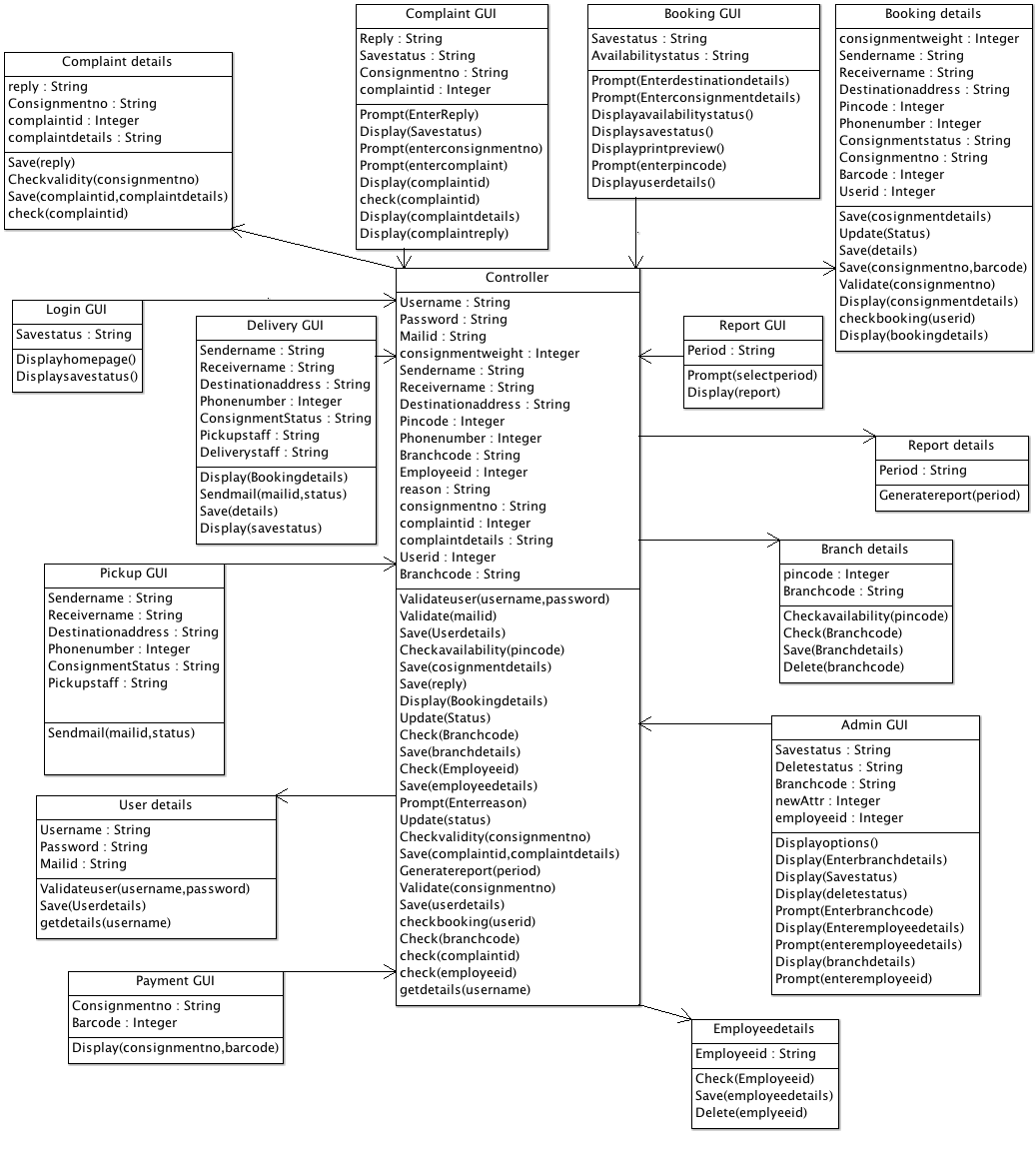
Enables the staff to take printout of barcode generated.





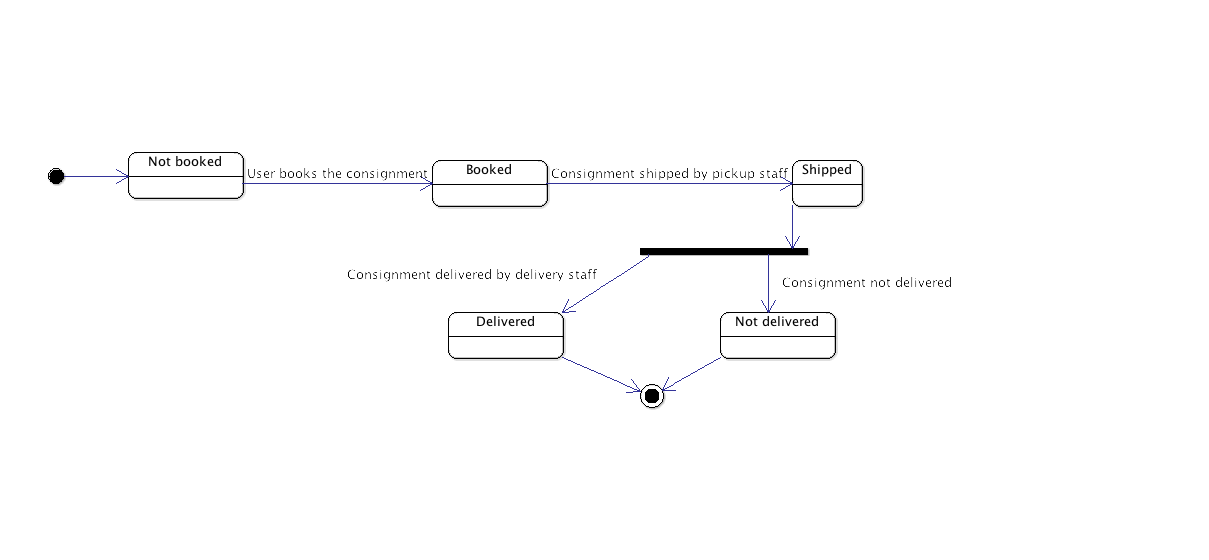
**Class Diagram:**

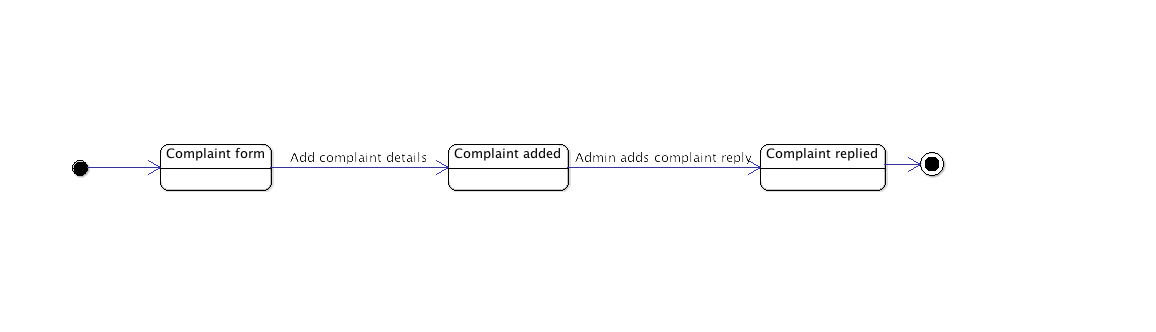
Class diagram in the [Unified Modeling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of static structure diagram that describes the structure of a system by showing the system's [classes](http://en.wikipedia.org/wiki/Class_(computer_science)), their attributes, operations (or methods), and the relationships among objects.



**State chart diagram:**

The state chart diagram depicts the behavior of the system over time. A state diagram is typically drawn for every class in the system that contains significant dynamic behavior. The behavior of the class is modeled in terms of what states it is in at various times, what actions it performs while in various states, and when it transitions from state to state based on events in the system.



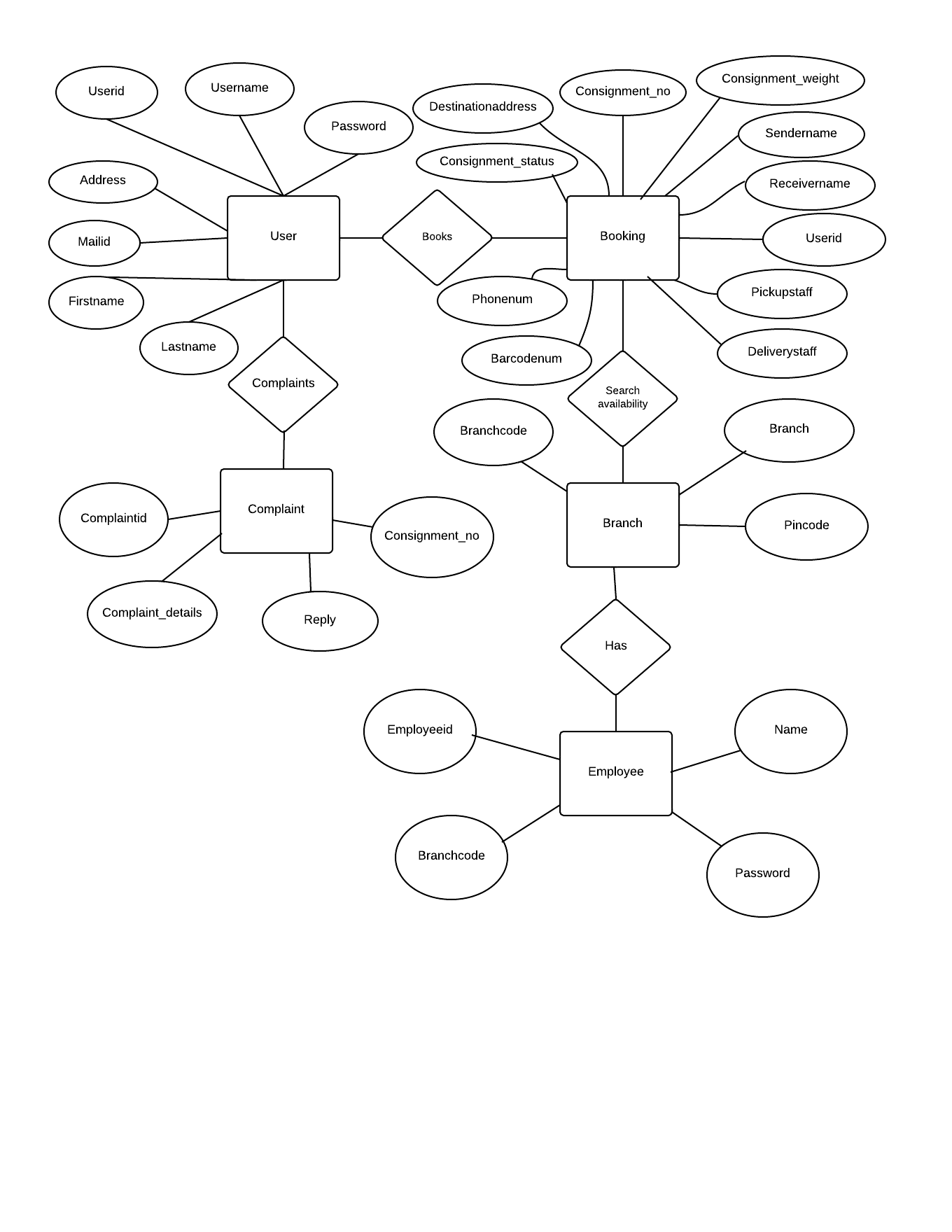


# Data Design

During detaileddesign, the internal logic of each of the modules specified in the system is decided. Thus, this part specifies the details of the data structures of each of the modules. The main tables used in CMS are User, Employee, Booking, Complaint, Branch.

## ER Diagram

An entity–relationship model (ER model) is a [data model](http://en.wikipedia.org/wiki/Data_modeling) for describing a [database](http://en.wikipedia.org/wiki/Database) in an abstract way. ER Modelling is based on the set of basic data entities in the problem domain and the relationship among entities.



## Data Description

TABLE 1

Name : User

Primary Key : Userid

Description : It includes the user details.

|  |  |  |
| --- | --- | --- |
| Fieldname | Datatype | Size |
| Userid | Integer |  |
| Username | Varchar | 12 |
| First\_name | Varchar | 12 |
| Last\_name | Varchar | 12 |
| Address | Varchar | 40 |
| Phone\_number | Integer | 10 |
| Password | Varchar | 25 |
| Mailid | Varchar | 25 |

TABLE 2

Name : Booking

Primary Key : Consignment\_no

Foreign Key : User\_id,Pickup\_staff,Delivery\_staffs

Description : It includes the booking details of the consignment.

|  |  |  |
| --- | --- | --- |
| Fieldname | Datatype | Size |
| Consignment\_no | Varchar | 10 |
| Consignment\_weight | Integer |  |
| User\_id | Integer |  |
| Sender\_name | Varchar | 24 |
| Receiver\_name | Varchar | 24 |
| Destination\_address | Varchar | 40 |
| Phone\_number | Integer |  |
| Consignment\_status | Varchar | 8 |
| Pickup\_staff | Varchar | 10 |
| Delivery\_staff | Varchar | 10 |
| Barcode\_no | Integer | 20 |

TABLE 3

Name : Branch

Primary Key : Branchcode

Description : It includes the branch details.

|  |  |  |
| --- | --- | --- |
| Fieldname | Datatype | Size |
| Branchcode | Varchar | 5 |
| Branch | Varchar | 20 |
| Pincode | Integer | 6 |

TABLE 4

Name : Employee

Primary Key : Employee\_id

Foreign Key : Branchcode

Description : It includes the employee details.

|  |  |  |
| --- | --- | --- |
| Fieldname | Datatype | Size |
| Employee\_id | Varchar | 8 |
| Name | Varchar | 25 |
| Branchcode | Varchar | 8 |
| Password | Varchar | 20 |

TABLE 5

Name : Complaint

Primary Key : Complaint\_id

Foreign Key : Consignment\_no

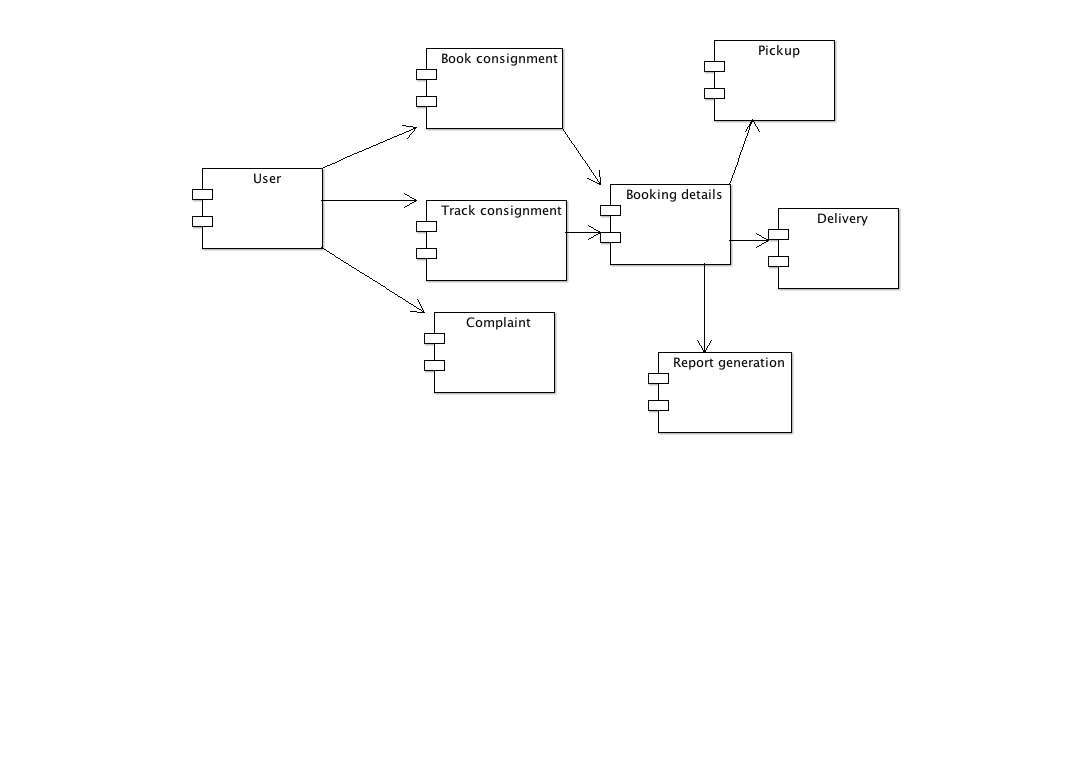
Description : It includes the complaint details.

|  |  |  |
| --- | --- | --- |
| Fieldname | Datatype | Size |
| Complaintid | Integer |  |
| Complaint\_details | Varchar | 100 |
| Reply | Varchar | 100 |
| Consignment\_no | Varchar | 10 |

# ****Component Design****

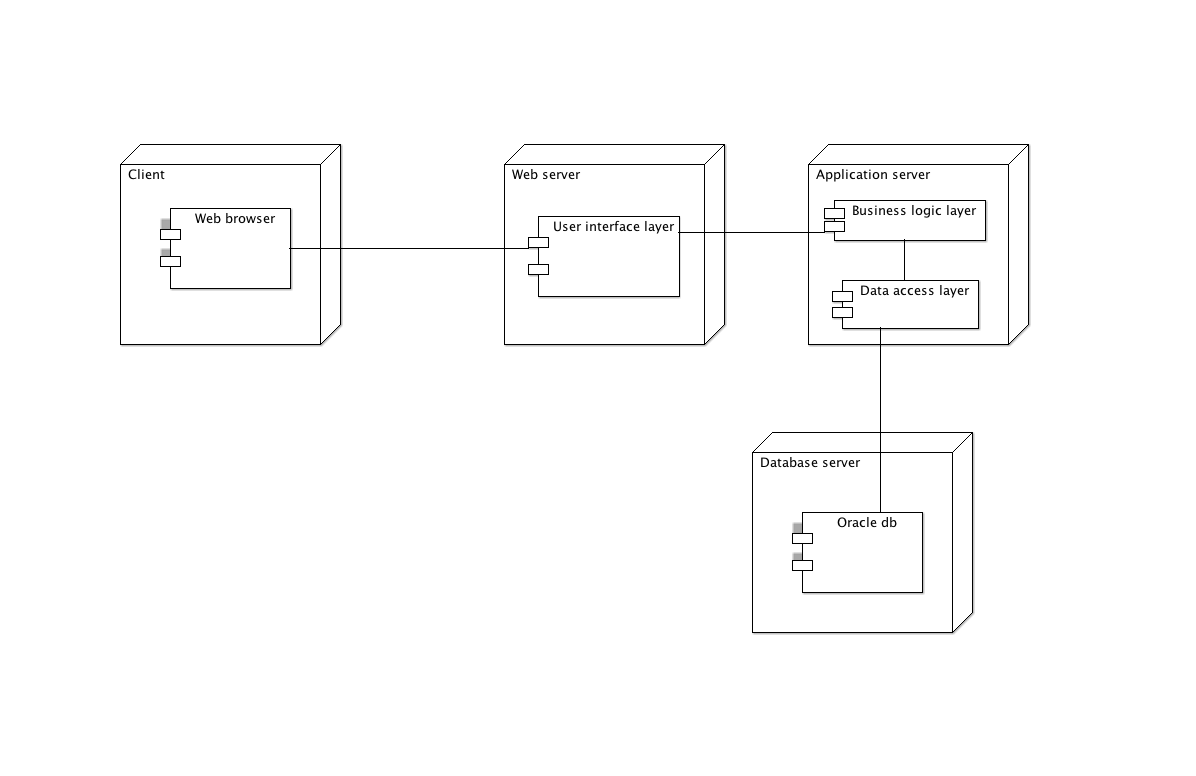
**Component Diagram**:

The component diagram's main purpose is to show the structural relationships between the components of a system,that is, depicts how [components](http://en.wikipedia.org/wiki/Component_(UML)) are wired together to form larger components and or [software systems](http://en.wikipedia.org/wiki/Software_system).



**Deployment Diagram:**

Shows the configuration of the processing nodes at run-time and the components that live on them.



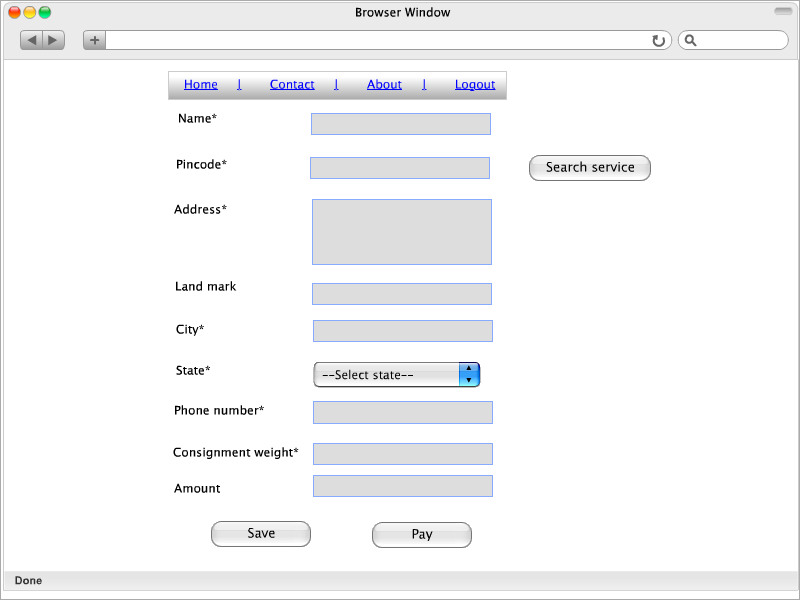
# ****Human Interface Design****

## Overview Of User Interface

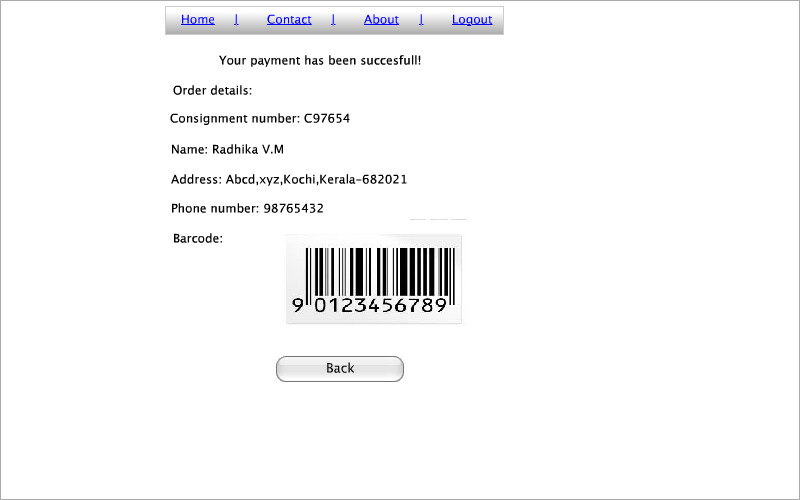
A user interface is the system by which people ([users](http://en.wikipedia.org/wiki/User_(computing))) [interact](http://en.wikipedia.org/wiki/Interaction) with a [machine](http://en.wikipedia.org/wiki/Machine). The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various [systems](http://en.wikipedia.org/wiki/System), and provide a means of input, allowing the users to manipulate a system and output, allowing the system to indicate the effects of the users' manipulation.

## Screen Images

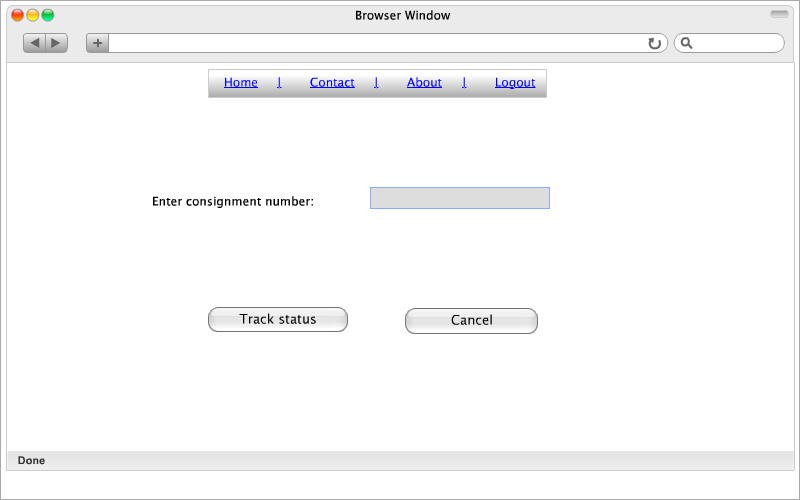
Booking

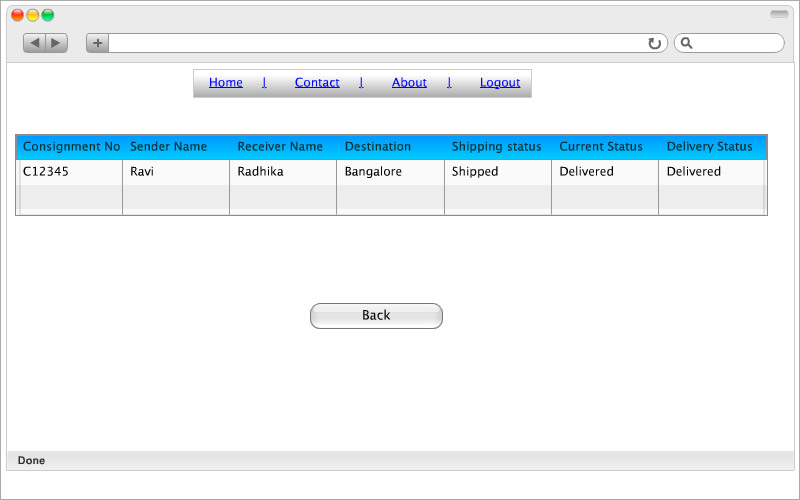


Confirmation

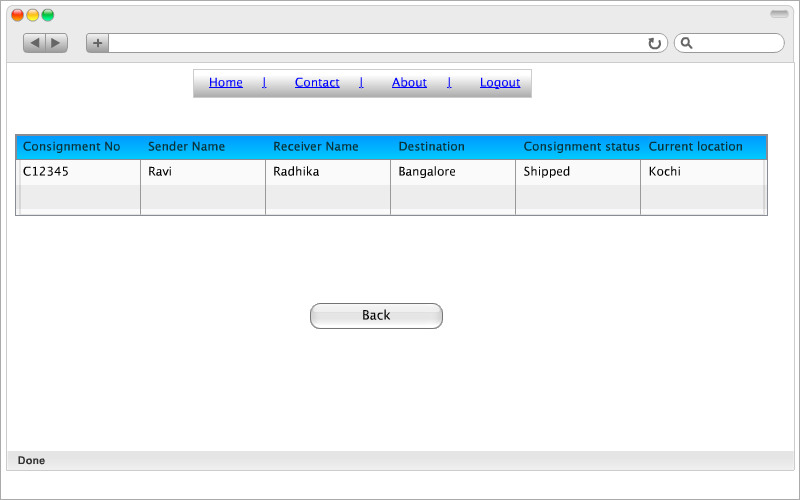


Track status

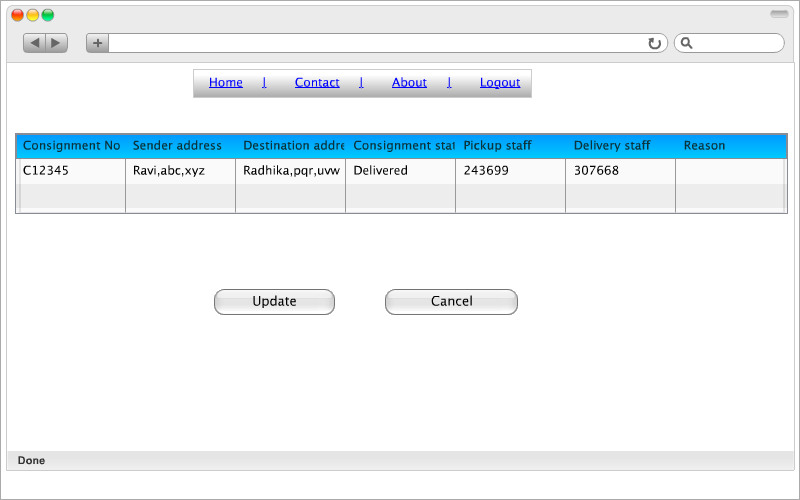




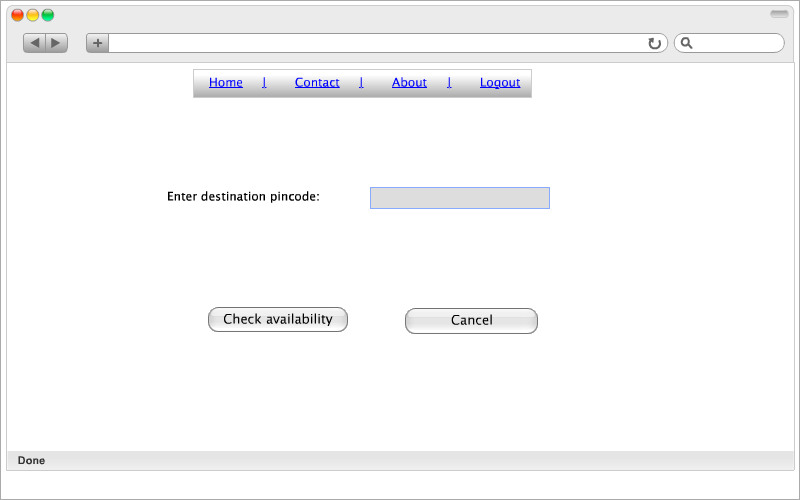
Pickup



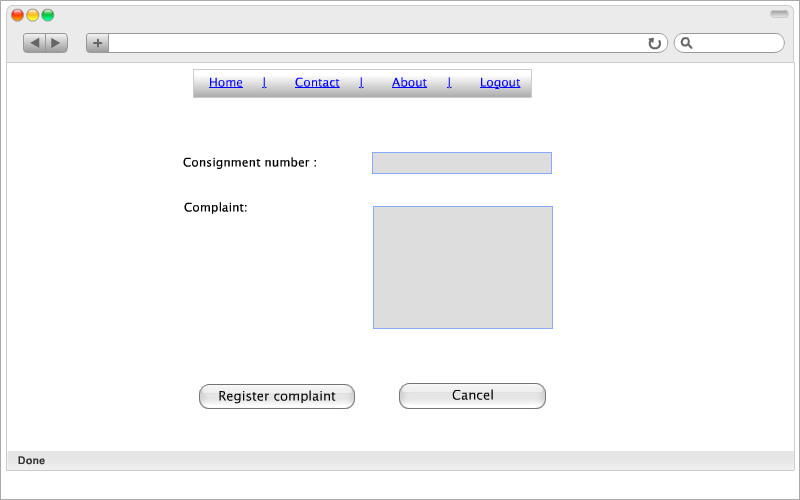
Delivery



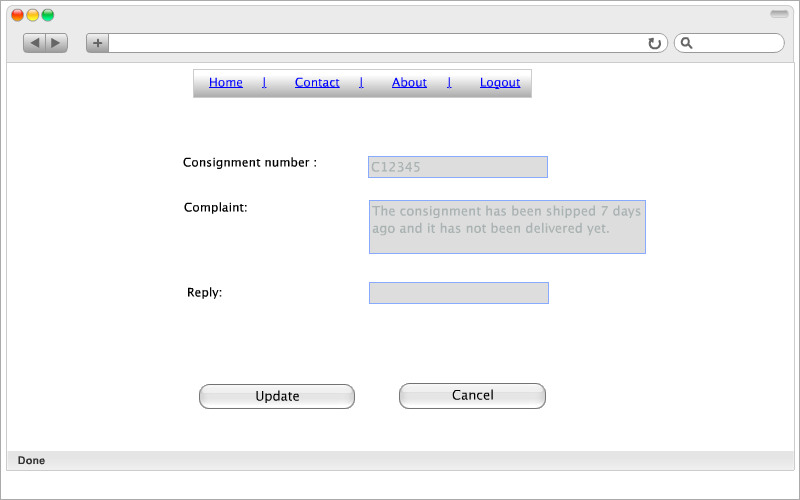
Search availability



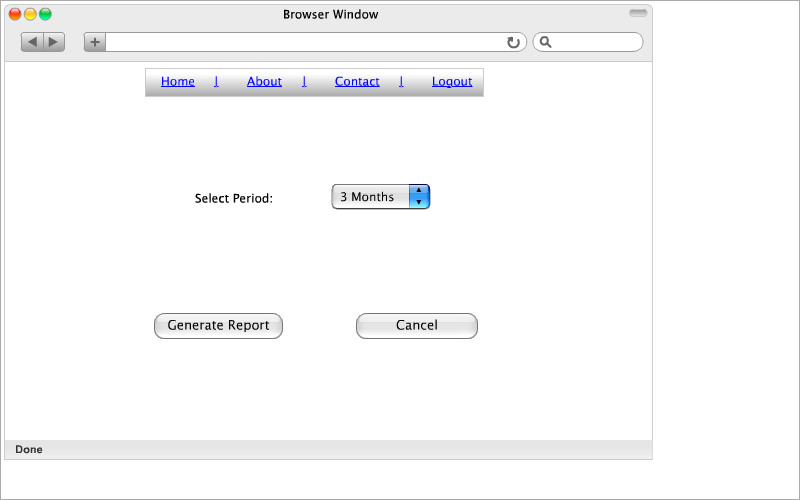
View complaint



Complaint reply



Report generation



# ****Requirement traceability matrix****

A traceability matrix is a document, usually in the form of a table, that correlates any two [baselined documents](http://en.wikipedia.org/wiki/Baseline_(configuration_management)) that require a many-to-many relationship to determine the completeness of the relationship. It is often used with high-level [requirements](http://en.wikipedia.org/wiki/Requirement) (these often consist of marketing requirements) and detailed requirements of the product to the matching parts of [high-level design](http://en.wikipedia.org/wiki/High-level_design), detailed design, [test plan](http://en.wikipedia.org/wiki/Test_plan), and [test cases](http://en.wikipedia.org/wiki/Test_case).

The requirement traceability matrix is usually developed in concurrence with the initial list of requirements. As the design specifications and test protocols are developed, the traceability matrix is updated to include the updated documents. The Requirements Traceability Matrix captures the complete user and system requirements for the system, or a portion of the system.

