****

**FPT UNIVERSITY**

**Taxi Caller Application on Windows Phone**

**F\_Taxi**

**Report #4 – Software Design Specification**

|  |  |
| --- | --- |
| F\_Taxi | |
| Group Member | SE02705 - Phạm Gia Hữu |
| SE02900 - Tạ Thiên Hưởng |
| SE02268 - Phạm Ngọc Hoàn |
| SE02314 - Nguyễn Văn Lập |
| Supervisor | Nguyễn Văn Sang |
| Project Code | F\_Taxi |

HaNoi, 11/2015

Table of Contents

[1 ARCHITECTURE DESIGN 3](#_Toc436999249)

[1.1 Choice of Architecture Design 3](#_Toc436999250)

[1.1.1 Layer Model overview 3](#_Toc436999251)

[1.1.2 Advantages and disadvantages of Layer Model 4](#_Toc436999252)

[1.1.3 The reason of choosing Layer Model 4](#_Toc436999253)

[1.2 Architectural Presentation 4](#_Toc436999254)

[1.3 Component/Package Design 4](#_Toc436999255)

[2 DETAILED DESIGN 4](#_Toc436999256)

[2.1 Common Design 4](#_Toc436999257)

[2.2 <UC xx- Use case name> 4](#_Toc436999258)

[2.2.1 Class Diagram 4](#_Toc436999259)

[2.2.2 Class Description 4](#_Toc436999260)

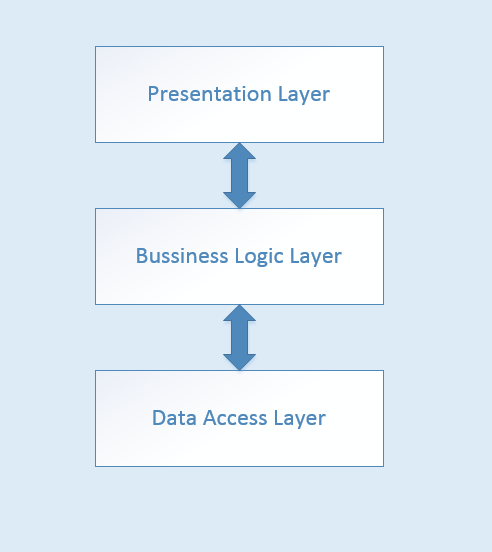
[2.2.3 Screen Design 6](#_Toc436999261)

[2.2.4 Sequence Diagram 6](#_Toc436999262)

# Architecture Design

## Choice of Architecture Design

### Layer Model overview



The components in Layer Model :

* Presentation Layer: are user interfaces, is the form of interactive program with system user. This layer missions communicate with end users to collect data and display the results, the data through the objects in the user interface.
* Business Logic Layer: assigned to are processing layer the business of the program as calculating, processing requests and check the validity and integrity of the data before it put on display on the screen or data processing before moving down data Data Access Layer to save data to the database.
* Data Access Layer: the function of this layer are communicate with the management system database

### Advantages and disadvantages of Layer Model

* Advantages :
  + The maintenance of your application is easier because of the low coupling between layers.
  + Adding more functionality to your application is made easier.
  + Layers make your application more testable.
* Disadvantages :
  + It is difficult to exactly assign of functionalities to the correct and appropriate layer
  + Negative impact to the performance of system

### The reason of choosing Layer Model

- Easily in develop and maintain system due to high flexibility, scalability and maintainability.

- Better support for test-driven development.

- It is good support for application built by project team has not too much developers thanks to the ease of reusability.

## Architectural Presentation

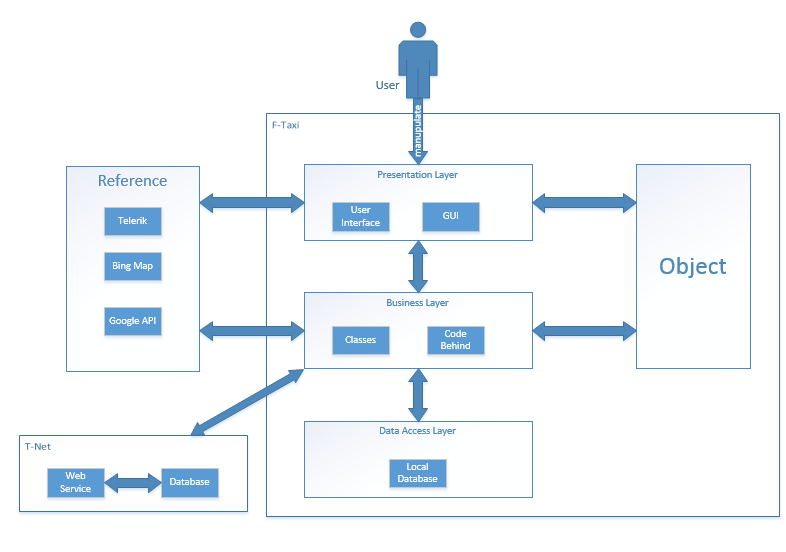


Figure 1.2: Overall System Diagram

In the F-Taxi application:

* Presentation Layer: using Telerik Controls to design some user interface. Bing map to display Maps and navigation with two main user. Google API. Notification to communicate with two kind of user Rider and Driver.
* Business Layer: using classes and object to work with T-Net, T-Net is a company supply web service and communicate with database.
* Data Access Layer: using local database.

## Component/Package Design

<Khái quát kiến trúc của hệ thống dưới góc nhìn về component - các thành phần biểu diễn bằng UML >

# Detail Design

## Common Design

<Phần mô tả thiết kế cho các thành phần chung của tất cả các chức năng>

## <UC xx- Use case name>

### Class Diagram

<Sơ đồ quan hệ giữa các lớp của chức năng>

### Class Description

<Phần mô tả chi tiết các lớp của chức năng>

#### <Class Name>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | <Class Name> | | | |
| **Description** |  | | | |
| **Base Class** |  | | | |
| **Constructor** |  | | | |
| **Prototype** |  | | | |
| **Source File** |  | | | |
| **Namespace** |  | | | |
| **Attributes** | Name | Type | Description | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |
|  |  |  |  |

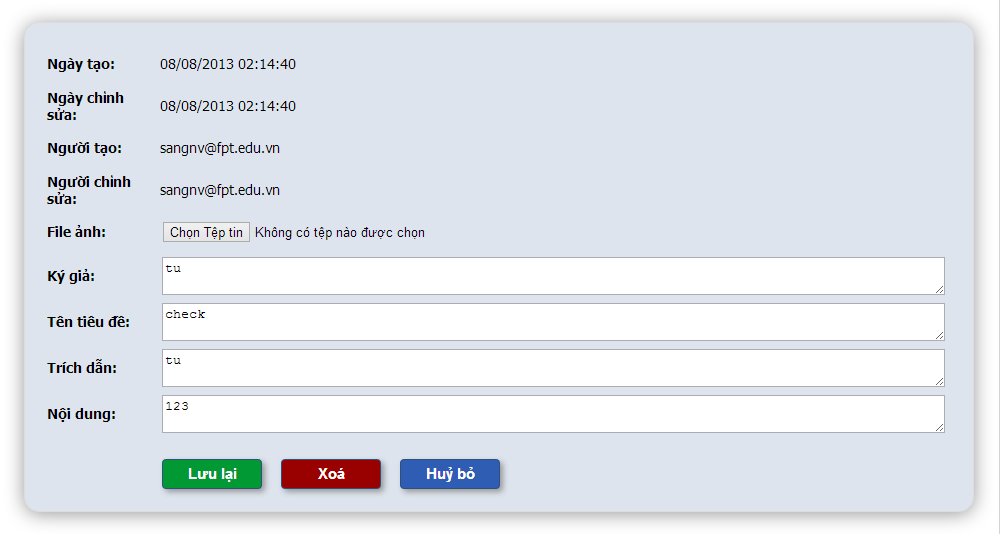
#### <Class Name>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | <Class Name> | | | |
| **Description** |  | | | |
| **Base Class** |  | | | |
| **Constructor** |  | | | |
| **Prototype** |  | | | |
| **Source File** |  | | | |
| **Namespace** |  | | | |
| **Attributes** | Name | Type | Description | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |
|  |  |  |  |

......

### Screen Design

<Screen Layout - Manage news- Delete news >



<Screen Definition>

**Table 4-x:** Manage news- Delete news

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Object/Control Name** | **Object/Control Name in English** | **Type** | **Required** | **Length** | **Description** |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |

### Sequence Diagram

#### <Use-case ID - Use case name>

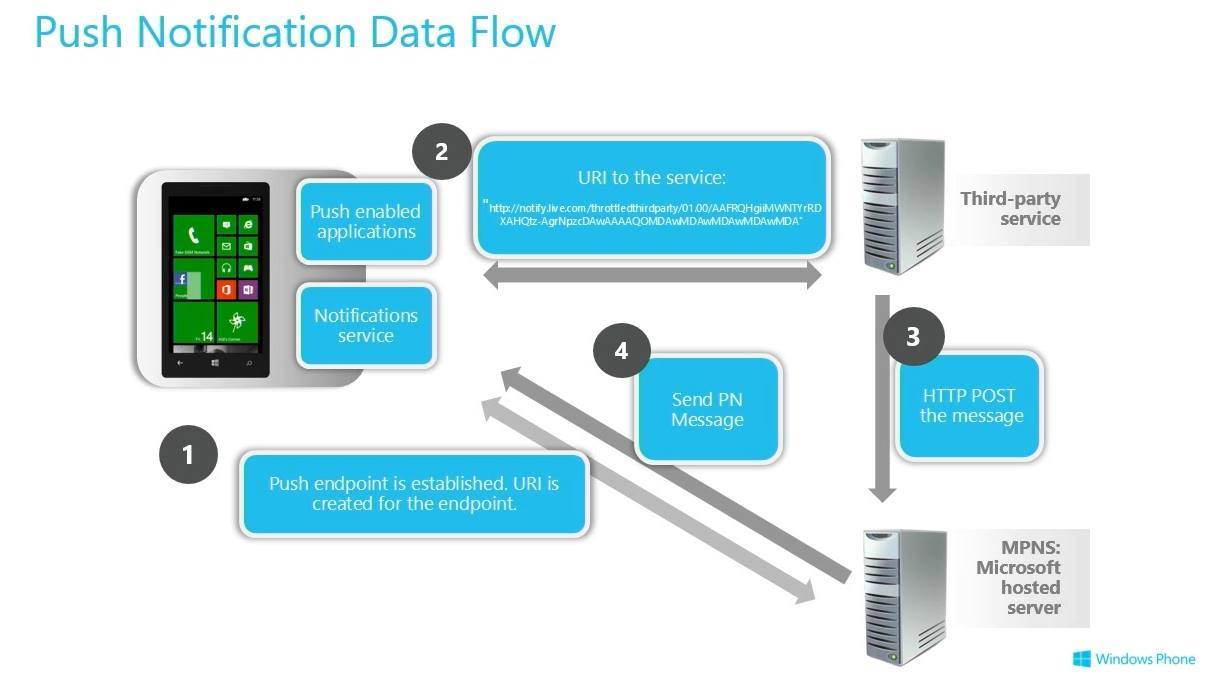
...

#### <Use-case ID - Use case name>

................

# Others

## Notification on Windows Phone



Description :

1. While running on the phone, the application creates an HttpNotificationChannel object, which provides a unique uniform resource identifier (URI) that uniquely identifies the phone where the application is installed and running.
2. The application then transmits the URI that identifies the phone to the entity that will initiate push notifications. In most cases, this entity is a web service created by the application developer.
3. The web service that initiates push notifications sends a specially crafted XML payload to the URI when a notification should be sent to the user phone. This delivers the notification to the Microsoft Push Notification Service for processing.
4. The Microsoft Push Notification Service detects when the user's phone has connectivity and sends pending notifications to the device. To optimize available bandwidth, these notifications might be sent in batches to the phone by the service.

# Reference

1. https://msdn.microsoft.com/en-us/library/hh221549.aspx