**SWE 599 Project, Fall 2015**

**Instructor: Fatih ALAGÖZ**

**Project Safe Send**

**Design Specifications Document**

**09.11.2015**

**Revision 1.0**

**By: Nuri Emre GÜRER**

Table of Contents

[Revision History 3](#_Toc434996238)

[1. Introduction 4](#_Toc434996239)

[2. System Design 4](#_Toc434996240)

[2.1. Software Architecture 4](#_Toc434996241)

[2.2 Software System Components 5](#_Toc434996242)

[2.3. Environment 6](#_Toc434996243)

[3. Detailed Design 6](#_Toc434996244)

[3.1. Web Service Classes 6](#_Toc434996245)

[3.2. Mobile Application Classes 10](#_Toc434996246)

[4. Database Design 14](#_Toc434996247)

[5. References 14](#_Toc434996248)

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Explanation** |
| 1.0 | 09.11.2015 | Initial design |

# 1. Introduction

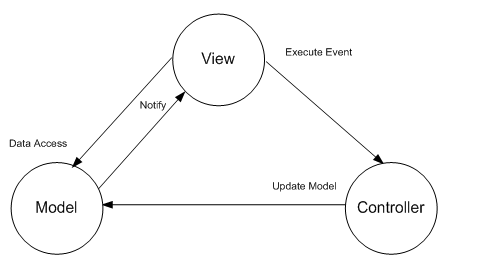
The purpose of this software project is to develop the Safe Send, safe and secure file transfer on iPhone devices, in Swift and iOS environment. Safe Send is a mobile application which allows its users to transfer photos or videos in a safe and secure manner. The application transfers the photos or videos with the desired encryption levels.

The design is based on Safe Send Requirements Specifications Document, final version. The notation used in this document to describe the design of the application is mainly UML and conforms to organizational specifications given in [2]. The software architecture, overall high-level structure, components in terms of packages and classes and design details of all application functions and the user interface are given in later sections of this document.

# 2. System Design

## 2.1. Software Architecture

MVC is used in Safe Send project as the software architectural pattern. View is the storyboard part on Swift project, Controller is the controller classes on Swift project and modal is web service methods. See the figure for MVC pattern.



## 2.2 Software System Components

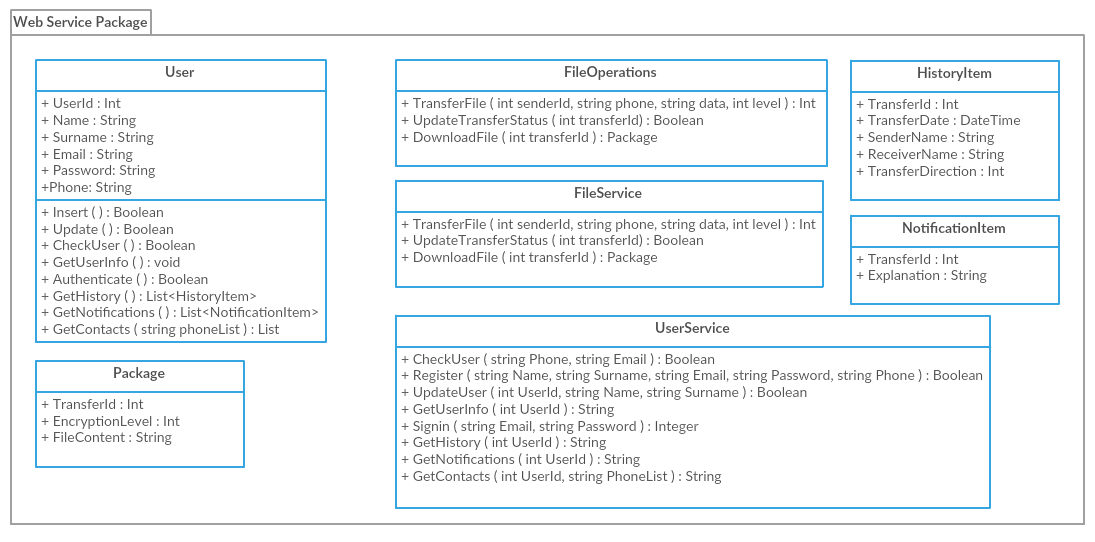
The Safe Send solution will comprise of following system components:

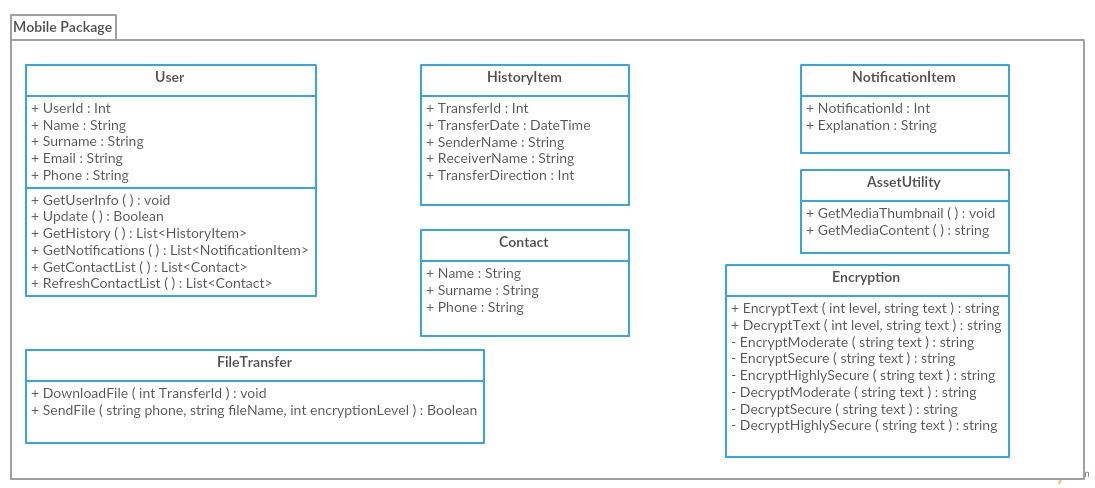
1. Mobile application
2. Web service application
3. Database

Mobile application constitutes the backbone of the project. User interacts with the mobile application. Web service application provides data for the mobile application.

Entity framework is used for the data access on the web service application.

The following diagrams shows the classes that the applications have.





## 2.3. Environment

Safe Send application development & runtime environment tools are listed below:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Version** |
| iOS | Operating System | 7.0 or later |
| Swift | Programming Language | 2 |
| XCode | IDE | 7 |
| MSSQL Server | Database | 2013 |
| C# | Programming Language |  |
| Entity Framework | Object-Relational Model | 6 |

# 3. Detailed Design

## 3.1. Web Service Classes

**3.1.1. UserService**

**Method CheckUser**

Input Parameters: Phone (String), Email (String)

Output Parameters: Boolean

Explanation: Calls the CheckUser method of the User object.

**Method Register**

Input Parameters: Name (String), Surname (String), Email (String), Password (String), Phone (String)

Output Parameters: Boolean

Explanation: Calls the Insert method of the User object.

**Method UpdateUser**

Input Parameters: UserId (Integer), Name (String), Surname (String)

Output Parameters: Boolean

Explanation: Calls the Update method of the User object.

**Method GetUserInfo**

Input Parameters: UserId (Integer)

Output Parameters: String

Explanation: Initializes the User object with the given UserId parameter and returns the json serialized User object.

**Method Signin**

Input Parameters: Email (String), Password (String)

Output Parameters: Integer

Explanation: Calls the Authenticate method of the User object.

**Method GetHistory**

Input Parameters: UserId (Integer)

Output Parameters: String

Explanation: Calls the GetHistory method of the User object and returns the json serialized list.

**Method GetNotifications**

Input Parameters: UserId (Integer)

Output Parameters: String

Explanation: Calls the GetNotifications method of the User object and returns the json serialized list.

**Method GetContacts**

Input Parameters: UserId (Integer), PhoneList (String)

Output Parameters: String

Explanation: Calls the GetContacts method of the User object.

**3.1.2. FileService**

**Method TransferFile**

Input Parameters: SenderId (Integer), Phone (String), Data (String), Level (Integer)

Output Parameters: Integer

Explanation: Calls the TransferFile method of FileOperations class.

**Method UpdateTransferStatus**

Input Parameters: TransferId (Integer)

Output Parameters: Boolean

Explanation: Calls the UpdateTransferStatus method of FileOperations class.

**Method DownloadFile**

Input Parameters: TransferId (Integer)

Output Parameters: String

Explanation: Calls the DownloadFile method of FileOperations class.

**3.1.3. User**

**Method Insert**

Input Parameters: No input parameters

Output Parameters: Boolean

Explanation: Inserts the user to the database with the object properties.

**Method Update**

Input Parameters: No input parameters

Output Parameters: Boolean

Explanation: Updates the information of the user.

**Method CheckUser**

Input Parameters: No input parameters

Output Parameters: Boolean

Explanation: Checks if the user exists on the database. Returns true if not exists.

**Method GetUserInfo**

Input Parameters: No input parameters

Output Parameters: No output parameters

Explanation: Gets the information of the user from the database.

**Method Authenticate**

Input Parameters: No input parameters

Output Parameters: Integer

Explanation: Checks the email and password of the user and returns userId of the user. If email and password does not match, then returns -1.

**Method GetHistory**

Input Parameters: No input parameters

Output Parameters: List of history object

Explanation: Gets the transfer history of the user from the database.

**Method GetNotifications**

Input Parameters: No input parameters

Output Parameters: List of notification object

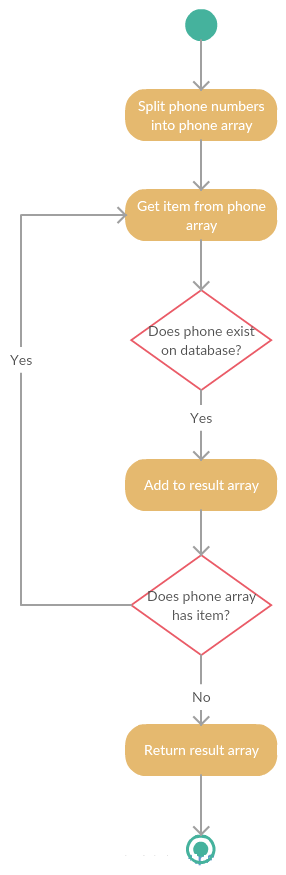
Explanation: Gets the list of files that the user did not receive.

**Method GetContacts**

Input Parameters: phoneList (String)

Output Parameters: String

Activity Diagram:



**3.1.4. FileOperations**

**Method TransferFile**

Input Parameters: senderId (integer), phone (string), data (string), level (integer)

Output Parameters: Integer

Explanation: Inserts the content of the transferred file to the database. Returns the transferId.

**Method UpdateTransferStatus**

Input Parameters: transferId (integer)

Output Parameters: Boolean

Explanation: Updates the status of the downloaded file on the database to completed status.

**Method DownloadFile**

Input Parameters: transferId (integer)

Output Parameters: Package object

Explanation: Gets the content of the transferred file from the database and returns the package.

## 3.2. Mobile Application Classes

**3.2.1. User**

**Method GetUserInfo**

Input Parameters: No input parameters

Output Parameters: No output parameters

Explanation: Calls the GetUserInfo method of the WCF service and sets the properties of the object.

**Method Update:**

Input Parameters: No input parameters

Output Parameters: Boolean

Explanation: Calls the Update method of the WCF service.

**Method GetHistory**

Input Parameters: No input parameters

Output Parameters: List<HistoryItem>

Explanation: Calls the GetHistory method of the WCF service.

**Method GetNotifications**

Input Parameters: No input parameters.

Output Parameters: List<NotificationItem>

Explanation: Calls the GetNotifications method of the WCF service.

**Method GetContactList**

Input Parameters: No input parameters.

Output Parameters: List<Contact>

Explanation: Gets the contact list from the storage of the mobile device.

**Method RefreshContactList**

Input Parameters: No input parameters.

Output Parameters: List<Contact>

Explanation: Calls the GetContacts method of the WCF service.

**3.2.2. FileTransfer**

Method DownloadFile

Input Parameters: TransferId (Integer)

Output Parameters: Boolean

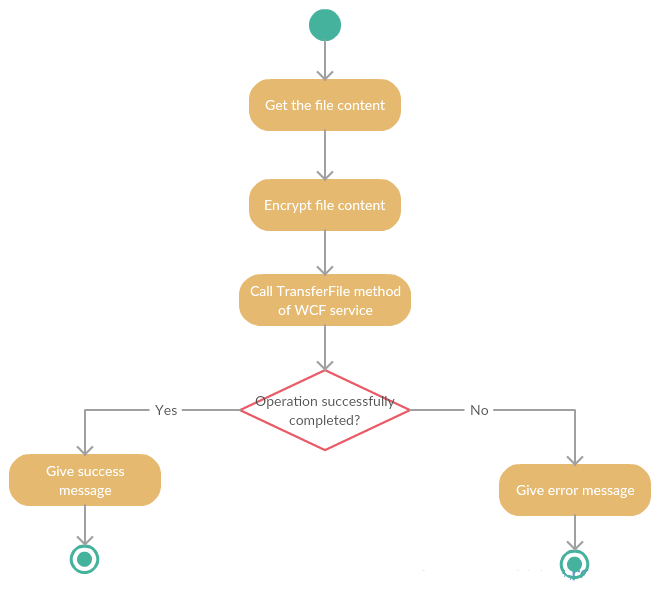
Explanation: Calls the DownloadFile method of WCF service and decrypts the data according to the encryption level and saves the data as image.

**Method SendFile**

Input Parameters: phone (string), (filename string), encryptionLevel (integer)

Output Parameters: Boolean

Activity Diagram:



**3.2.3. AssetUtility**

**Method GetMediaThumbnail**

Input Parameters: No input parameters

Output Parameters: No output parameters

Explanation: Gets the list of media files stored in the storage of the mobile device and shows as thumbnail icon on the screen.

**Method GetMediaContent**

Input Parameters: filename (String)

Output Parameters: String

Explanation: Gets the content of the selected file and converts it to string.

**3.2.4. Encryption**

**Method EncryptText**

Input Parameters: level (integer), text (string)

Output Parameters: string

Explanation: Calls one of the encryption methods according to the level and returns the result.

**Method DecryptText**

Input Parameters: level (integer), text (string)

Output Parameters: string

Explanation: Calls one of the decryption methods according to the level and returns the result.

**Method EncryptModerate**

Input Parameters: text (string)

Output Parameters: string

Explanation: Calls the encryption method of CryptoSwift library with AES-128 parameter.

**Method EncryptSecure**

Input Parameters: text (string)

Output Parameters: string

Explanation: Calls the encryption method of CryptoSwift library with AES-192 parameter.

**Method EncryptHighlySecure**

Input Parameters: text (string)

Output Parameters: string

Explanation: Calls the encryption method of CryptoSwift library with AES-256 parameter.

**Method DecryptModerate**

Input Parameters: text (string)

Output Parameters: string

Explanation: Calls the decryption method of CryptoSwift library with AES-128 parameter.

**Method DecryptSecure**

Input Parameters: text (string)

Output Parameters: string

Explanation: Calls the decryption method of CryptoSwift library with AES-192 parameter.

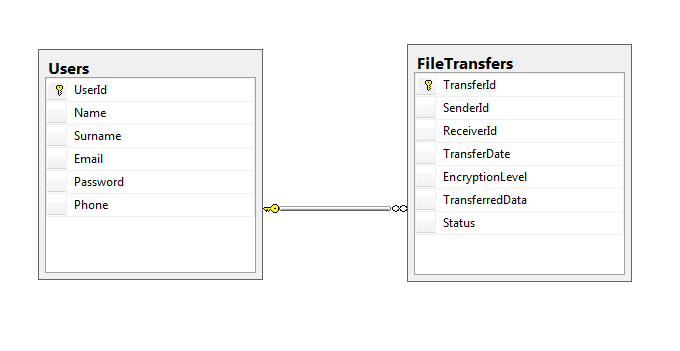
**Method DecryptHighlySecure**

Input Parameters: text (string)

Output Parameters: string

Explanation: Calls the decryption method of CryptoSwift library with AES-256 parameter.

# 4. Database Design



# 5. References

[1] Bennett-Object-oriented System Analysis and Design using UML 4thE

[2] UML2.0 Specifications, [http://www.uml.org/#UML2.0.](http://www.uml.org/#UML2.0)

[3] Safe Send Requirements Specifications Document v1.0