System/Solution Requirements

Specification

(SRS)

Online Charging

System

**2024**

Table of Contents

[1. Document Overview 4](#_Toc174608445)

[1.2. Scope&Purpose 4](#_Toc174608446)

[1.3. Version Numbers & Standards 4](#_Toc174608447)

[1.4. Audience 4](#_Toc174608448)

[1.5. Document Structure and Special Conventions 5](#_Toc174608449)

[1.6. Definitions, Symbols and Abbreviations 5](#_Toc174608450)

[2. Assumptions 6](#_Toc174608451)

[3. General Requirements 6](#_Toc174608452)

[3.2. Business Requirements 6](#_Toc174608453)

[3.3. User Requirements 6](#_Toc174608454)

[4. Functional Requirements 7](#_Toc174608455)

[4.2. Login Page (Android/IOS) 7](#_Toc174608456)

[4.3. Sign Up Page (Android/iOS) 7](#_Toc174608457)

[4.4. New Password Page (Android/iOS) 7](#_Toc174608458)

[4.5. Remaining Usage Page (Android/iOS) 8](#_Toc174608459)

[4.6. Login Page (Web/Desktop) 8](#_Toc174608460)

[4.7. Register Page (Web/Desktop) 8](#_Toc174608461)

[4.8. New Password Page (Web/Desktop) 8](#_Toc174608462)

[4.9. Remaining Usage Page (Web/Desktop) 9](#_Toc174608463)

[4.10. SMS Channel 9](#_Toc174608464)

[4.11. AOM 9](#_Toc174608465)

[4.12. Apache Kafka 10](#_Toc174608466)

[4.13. ABMF (Account&Balance Function) 10](#_Toc174608467)

[4.14. CGF (Charging GW Function) 10](#_Toc174608468)

[4.15. Oracle 10](#_Toc174608469)

[4.16. CHF (Charging Function) 10](#_Toc174608470)

[4.17. VoltDB 11](#_Toc174608471)

[4.18. TGF (Traffic Generator) 11](#_Toc174608472)

[4.19. Hazelcast 11](#_Toc174608473)

[4.20. Test 1 11](#_Toc174608474)

[4.21. Test 2 11](#_Toc174608475)

[4.22. Test 3 (Android App) 12](#_Toc174608476)

[4.23. Test 4 (IOS App) 12](#_Toc174608477)

[4.24. Test 5 (Web Page) 12](#_Toc174608478)

[4.25. Test 6 12](#_Toc174608479)

[5. Architecture of the Project 13](#_Toc174608480)

[6. Solution Overview 14](#_Toc174608481)

[6.2. Mobile Solution Overview 14](#_Toc174608482)

[6.3. Web Solution Overview 16](#_Toc174608483)

[6.4. Real-time Solution Overview 18](#_Toc174608484)

[6.5. Database Solution Overview 19](#_Toc174608485)

[6.6. High-Level Design Details 19](#_Toc174608486)

[7. Diagrams 21](#_Toc174608487)

[8. End to End User Cases 23](#_Toc174608488)

[8.2. Use Case – 1 23](#_Toc174608489)

[8.3. User Case – 2 24](#_Toc174608490)

[8.4. User Case - 3 25](#_Toc174608491)

[8.5. Use Case – 4 25](#_Toc174608492)

[8.6. Use Case – 5 26](#_Toc174608493)

[8.7. Use Case – 6 27](#_Toc174608494)

[8.8. Use Case – 7 27](#_Toc174608495)

[9. Appendix 28](#_Toc174608496)

[9.2. Document Control 28](#_Toc174608497)

[9.3. Change Control § Distribution 28](#_Toc174608498)

[9.4. Version History 28](#_Toc174608499)

[9.5. Approvals 28](#_Toc174608500)

# Document Overview

## Scope&Purpose

The scope of this document is to provide system/solution requirements specification (SRS) tool for capturing of the System/Solution related to Account, CRM, OM and MW requirements for “DataDito” project.

## Version Numbers & Standards

This Specification has been produced by the i2i Systems (i2i) following 3GPP specifications alike

versioning scheme and follows the recommendations of “IEEE Std 830-1998, IEEE Recommended

Practice for Software Requirements Specifications” standard.

The contents of the present document are subject to continuing work and may be changed by i2i.

Should the i2i modify the contents of the present document, it will be re-released with an identifying

change of **release date** and an increase in **version number** as follows:

Version x.y.z

**where:**

**x** **the first digit:**

1 presented for information;

2 presented for approval;

3 or greater indicates approved document under change control.

**y** **the second digit** is incremented for all changes of substance, i.e. technical enhancements,

corrections, updates, etc.

**z** **the third digit** is incremented when editorial only changes have been incorporated in the document.

## Audience

The document may be read by the below audience;

• Business/System Analysts

• Solution/System Architects

## Document Structure and Special Conventions

* Font: Times New Roman, Size 12, Size 24 for headers
* Numbering: Lists and dots are used for numbering

## Definitions, Symbols and Abbreviations

|  |  |  |
| --- | --- | --- |
| **TERM** | **DESCRIPTION** |  |
| OCS | Online Charging Solution |
| API | Application Programming Interface |
|  |  |
|  |  |
| BSS | Business Support System |
| DB | Database |
| GUI | Graphical User Interface |
| SAS | System/Solution Architectural Specification |
| SRS | System/Solution requirements Specification |
| CDR | Call Data Record |
| WA | Workaround |
| SMS | Short Message Service |
| MMS | Multimedia Messaging Service |
| FTP | File Transfer Protocol |
| CCR | Credit Control Request |
| CCA | Credit Control Answer |
| 3GPP | Third Generation Partnership Program |
| UDR | Usage Data Record |
| MSISDN | Mobile Station International Subscriber Directory Number |
| IMSI | International Mobile Subscriber Identity |
| WS | Web Service |
| SOI | Service Oriented Interface |
| CMS | Content Management System |
| CRM | Customer Relations Management |
| OM | Order Management |
|  |  |

# Assumptions

1. Relationships established with operators and other network service providers will be seamlessly and timely integrated.
2. Subscribers' devices and applications used will be compatible with and support the online billing system.
3. The necessary encryption and security measures for the security of online billing transactions will be successfully implemented.
4. Required data integrations between source systems will be carried out smoothly.
5. Subscribers' service usage and payment behaviors will align with expectations.
6. The project will comply with relevant legal regulations and regulatory requirements.
7. The online billing system will have the capacity and performance to meet current user demands and workloads.

# General Requirements

## Business Requirements

1. The system must calculate accurate charges based on real-time service usage by users and provide instant notifications.
2. The system must address any billing errors and provide information for corrections.
3. The system must have the ability to integrate with other enterprise systems, such as customer relationship management and billing systems.
4. The system must implement necessary security measures to protect user data and facilitate secure payment transactions.

## User Requirements

1. Users should have a general understanding of technology, particularly in terms of proficiency in performing online transactions.
2. Users must have the ability to manage their accounts, view bills, and make payments.
3. Users should understand how they use online services, including preferences and frequency of use.
4. Users should know their preferred payment methods, such as credit cards, bank transfers, e-wallets, etc.
5. Users have expectations regarding the security of their payment transactions and personal information.
6. Users may be accustomed to or prefer using services via mobile devices.
7. Users may need to resolve queries or seek assistance through customer support services.
8. Users are sensitive about how their personal data is processed and protected.

# Functional Requirements

## Login Page (Android/IOS)

* When the application is tapped, it should prompt the user for their phone number and password.
* At the bottom on the middle side, there should be a "Login" button that allows entry if the correct password and phone number are entered.
* To the top of the Login button, there should be a "Register" button.
* Top of the Register button, there should be a "Forgot Password" button.
* Below the text boxes which are for phone number and password, there is a “Remember me” Button to save your credentials for next time

## Register (Android/iOS)

* After clicking the "Register" button on the login page, the user should be directed to this page.
* Users should be able to register using their First Name, Last Name, National ID, Phone Number, Email, and Password.
* User also must select predetermined package from a box, “Tariff”.
* A "Register" button should be present.
* After pressing the "Register" button, if the required information is entered correctly, the user should be redirected to the login page.
* Users can go back to “Login” page by clicking “Go back” button.
* Turkish characters will be supported.

## New Password Page (Android/iOS)

* Users should be asked for their National ID and Email information.
* Below the information fields, there should be a "Send Recovery Email" button.
* If the information is correct, the user should get an email for password
* Below the section where the new password is requested, there should be a " Send Recovery Email " button.
* After pressing the " Send Recovery Email " button, the user should be redirected to the login page.
* Users can go back to “Login Page” by clicking “Go back” button.

## Remaining Usage Page (Android/iOS)

* On this page, the user's remaining usage should be displayed in a horizontal chart for mobile applications.
* At the bottom of the page, there should be a "Logout" button to return to the “Login page”

## Login Page (Web/Desktop)

* When the Web page is opened, it should prompt the user for their phone number and password.
* At the bottom of the page, on the middle side, there should be a "Login" button that allows entry if the correct password and phone number are entered.
* Below the text boxes, there is a “Remember me” Button to save your credentials for next time
* To the bottom of the “Remember me” button, there should be a "Forgot Password" button.
* Below the "Forgot Password" button, there should be a "Register" button.

## Register Page (Web/Desktop)

* After clicking the "Sign Up" button on the login page, the user should be directed to this page.
* Users should be able to register using their First Name, Last Name, National ID, Phone Number, Email and Password.
* Additionally, the user must select predetermined package using a combo box at the bottom. The predefined packages are described in the following pages
* A "Register" button should be present at the bottom.
* After pressing the "Register" button, if the required information is entered correctly, the user should be redirected to the login page.
* Turkish characters will be supported.

## New Password Page (Web/Desktop)

* Users should be asked for their National ID and Email information.
* Below the information fields, there should be a "Send Recovery Email" button.
* If the information is correct, the user should get an email for password
* Below the section where the new password is requested, there should be a " Send Recovery Email " button.
* After pressing the " Send Recovery Email " button, the user should be redirected to the login page.

## Remaining Usage Page (Web/Desktop)

* On this page, the user's remaining usage should be displayed in a horizontal chart.
* At the bottom of the page, there should be a "Logout" button to return to the “Login page”
* There is an image of a network beneath the logo to enhance the visuality.

## SMS Channel

* The remaining usage in the SMS channel will be sent to the user's phone as an SMS by contacting MW by typing "4545", " ", "KALAN".
* Incoming message in SMS channel must support Turkish characters.

## AOM

* MW will fetch usage or remaining query messages from CMR and send to VoltDB
* CMR will send the message from VoltDB.
* MW will send the remaining query message to VoltDB and retrieve the message from VoltDB.
* It will send the feedback to MW whether the submitted request is true or false.
* MW will receive data from SF.
* The remaining tariff amounts, user information and user tariffs from MW SF will be sent to Oracle.
* MW Oracle configurations and data insertion function will be written.
* MW Oracle login, membership and balance will be created.
* MW will send balance update transactions to Oracle.
* MW will send an e-mail when the user forgets her password.

## Apache Kafka

* Kafka will be run on Google Cloud.
* Kafdrop will be installed and run.
* Topics will be determined and added in Kafka.
* It will be connected to Kafka SF.
* Connection will be established to get user data and usage data from Kafka

OCS.

## ABMF (Account&Balance Function)

* SF will store new user information, tariff usage and user information from Kafka.
* SF will send this stored data to MW.

## CGF (Charging GW Function)

* CGF will store usage transactions from Kafka.
* CGF will send this stored data to MW.

## Oracle

* User information, tariff usage, new user information and tariff information coming from Oracle MW will be stored in the database.
* According to the message from Oracle Mw, the database will be updated.

## CHF (Charging Function)

* As OCS subscribers use internet, SMS and minutes, the amount used will be deducted from the subscriber's balance.
* OCS will manage the tariffs of subscribers. In tariff management, it will determine the applicable tariffs and pricing for certain services.
* OCS will send the tariffs it has determined and the amount of SMS, internet and minutes that should be reduced from the user's account as a voltdb message.

## VoltDB

* Updated tariff and used SMS, minutes and internet from VoltDB OCS will be updated in its own database.
* The user from VoltDB MW will receive the request for the remaining tariffs and will find this data and send it back to MW as a message.

## TGF (Traffic Generator)

* An HTTP server will be created over Google Cloud.
* DGW will receive random requests via Hazelcast.
* DGW will send requests to OCS.

## Hazelcast

* Hazelcast virtual machine will be installed over Google cloud.
* It will be possible to access the Hazelcast map from remote servers.
* Necessary libraries for Hazelcast map will be created for MW.
* Necessary libraries will be created for the Hazelcast map for the simulator.
* The simulator will send requests to DGW at regular intervals.
* The simulator will randomly generate data, SMS and minute data.
* The simulator will send the random requests it generates to the hazelcast map.

## Test 1

* ABMF
* CGF
* Notification

## Test 2

* VoltDB
* Hazelcast
* OracleDB
* Kafka
* AOM

## Test 3 (Android App)

* Resources : Java, Unix, SoapUI, Katalon, Postman, Selenium
* Use an automated test framework in order to test Android app.
* Create test cases document.

## Test 4 (IOS App)

* Resources : Java, Unix, SoapUI, Katalon, Postman, Selenium
* Use an automated test framework in order to test IOS app.
* Create test cases document.

## Test 5 (Web Page)

* Resources : Java, Unix, SoapUI, Katalon, Postman, Selenium
* Use an automated test framework in order to test Web page.
* Create test cases document.

## Test 6

* CHF

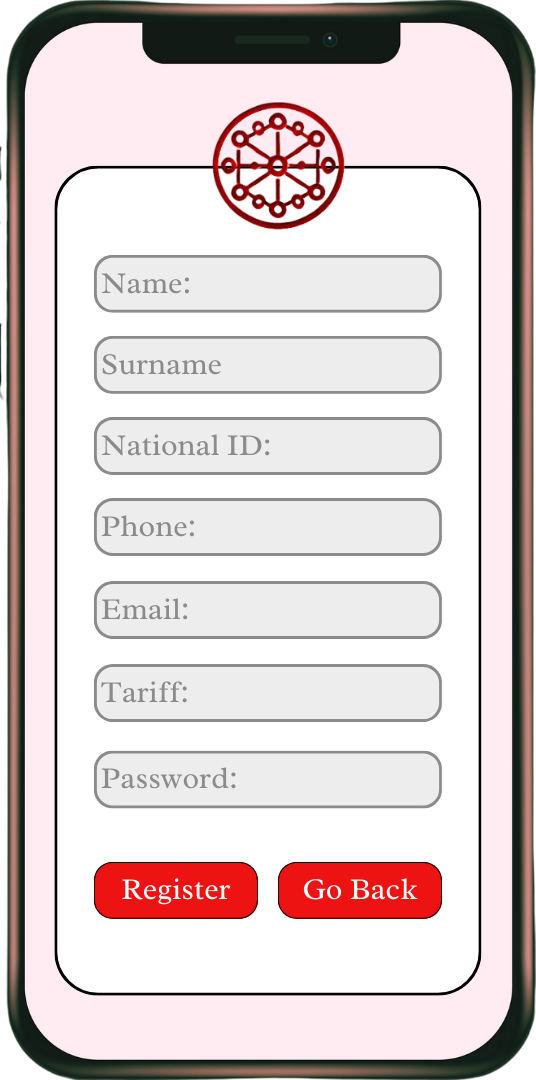
# Architecture of the Project

A computer screen shot of a diagram

Description automatically generated

# Solution Overview

## Mobile Solution Overview

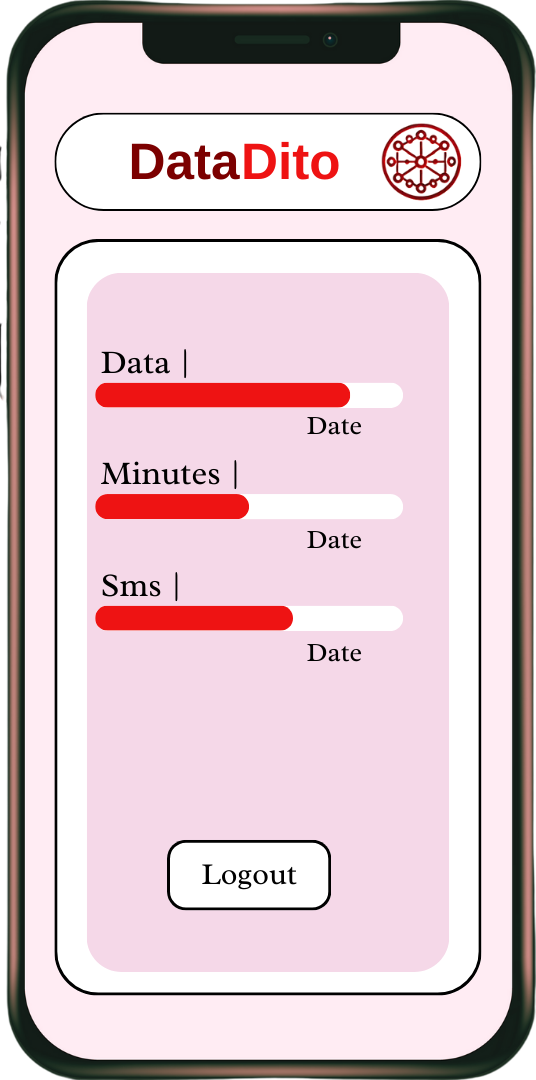
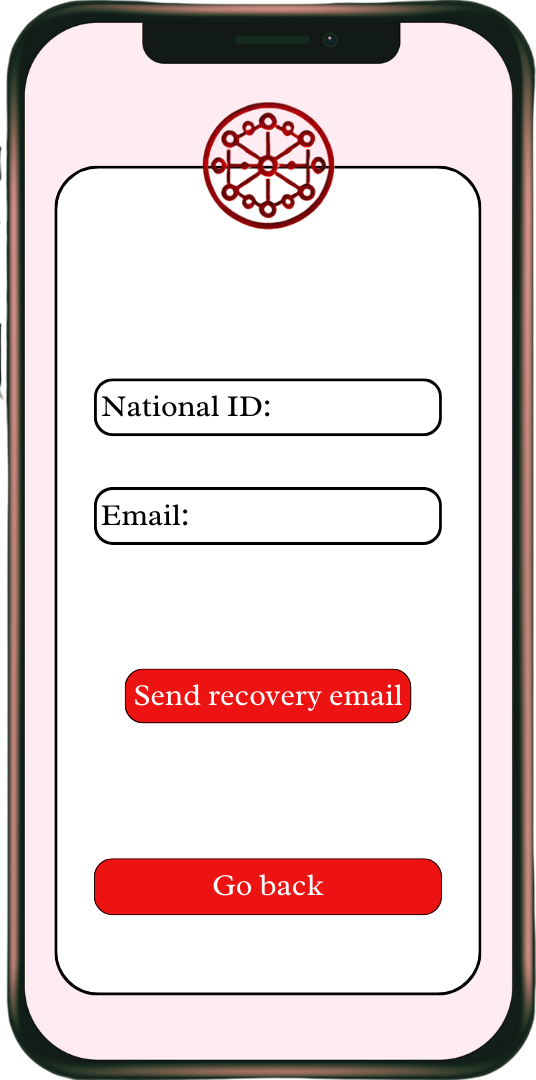
 

**Figure.2** Register

(Mobile)

**Figure.1** Login

(Mobile)

**Figure.4** Forgot Password (Mobile)

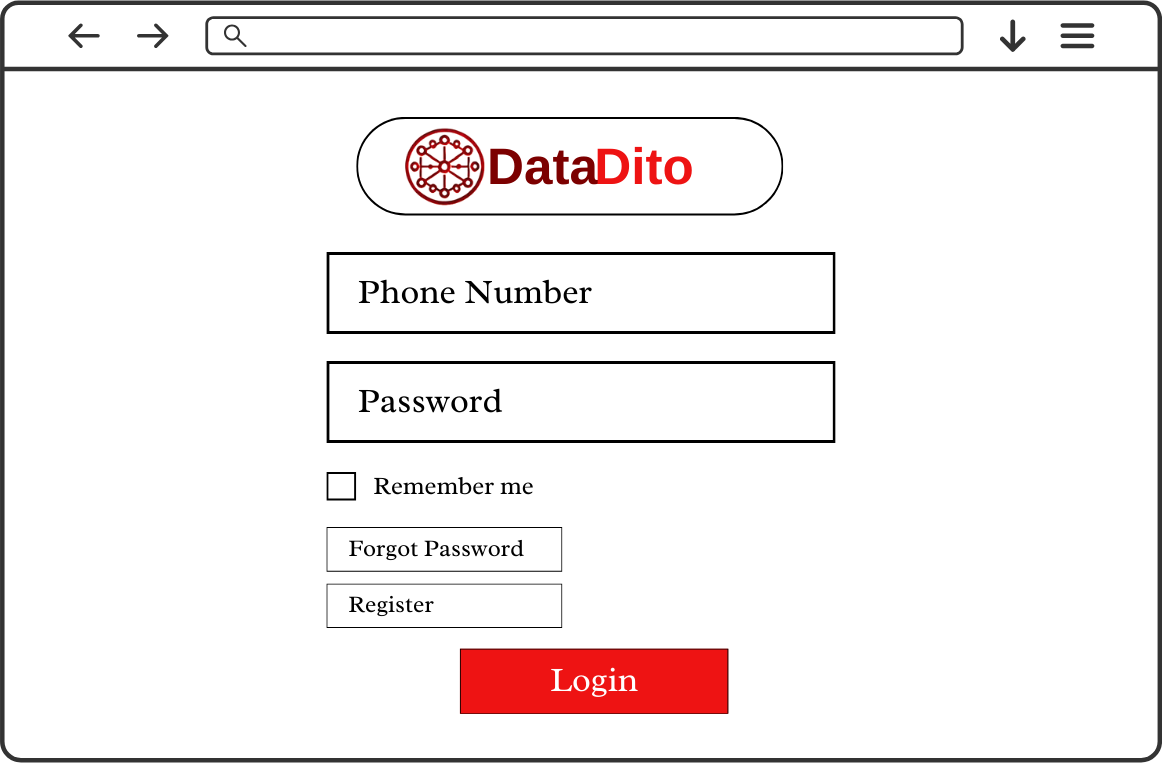
**Figure.3** User Interface (Mobile)

## Web Solution Overview

**Figure.5**

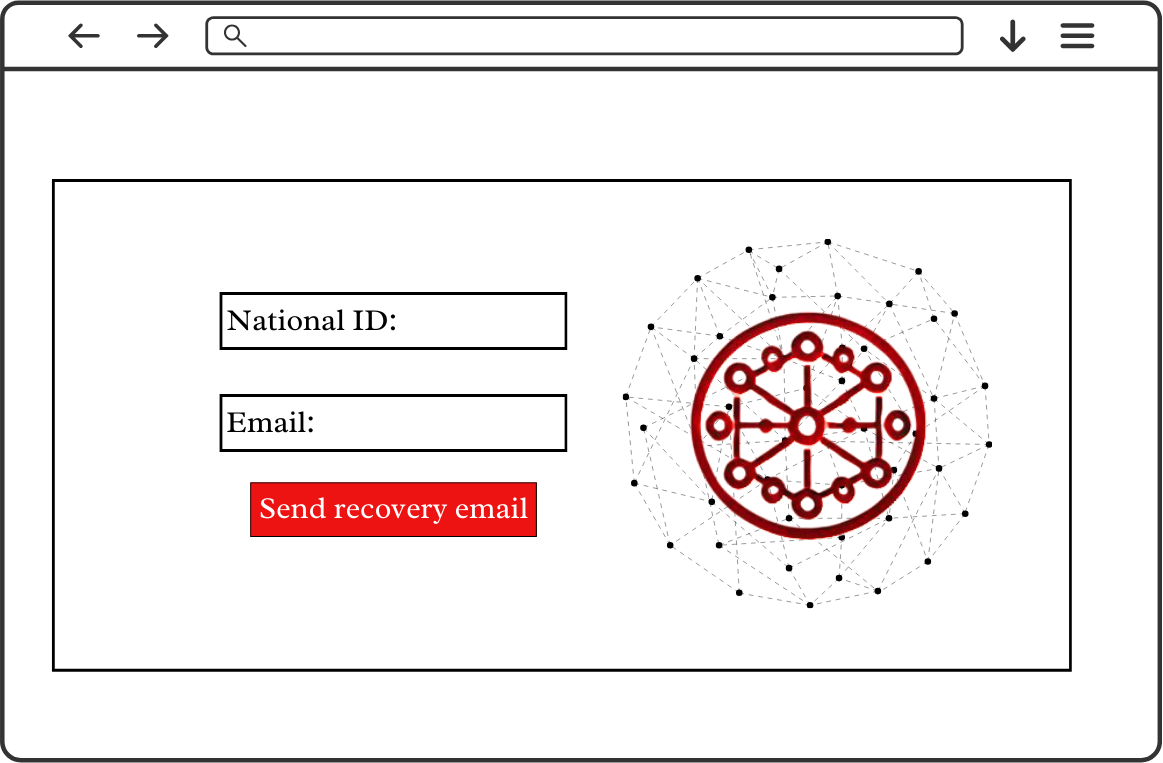
Login Page

(Web)



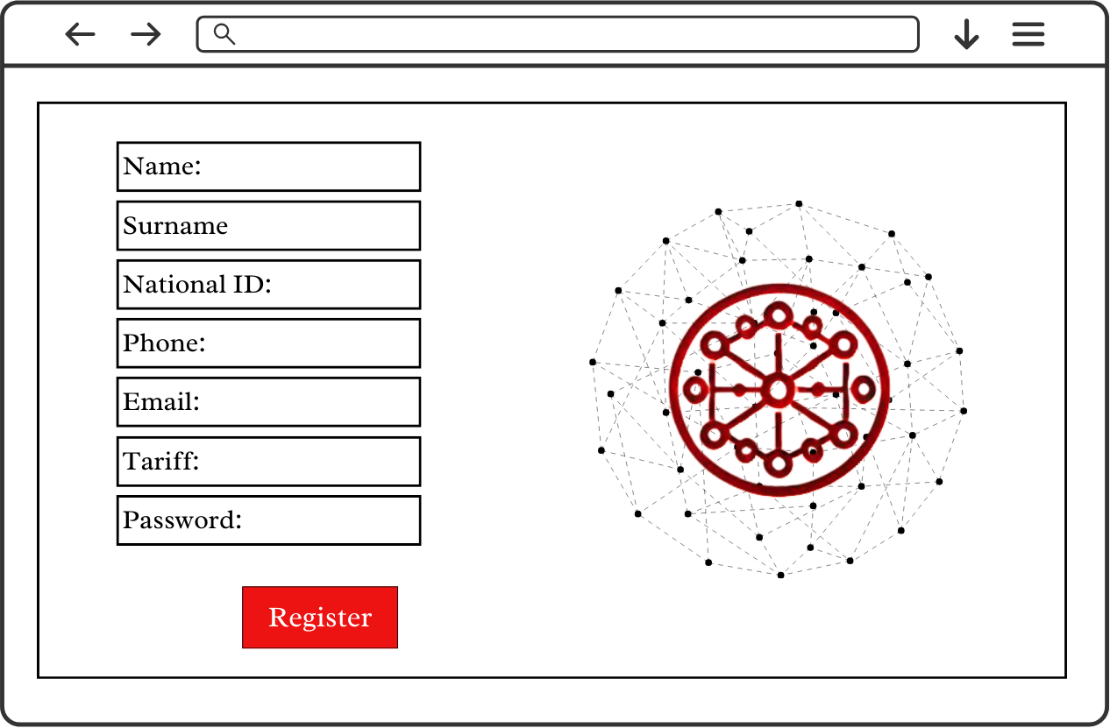
**Figure.6** Forgot Password Page

(Web)



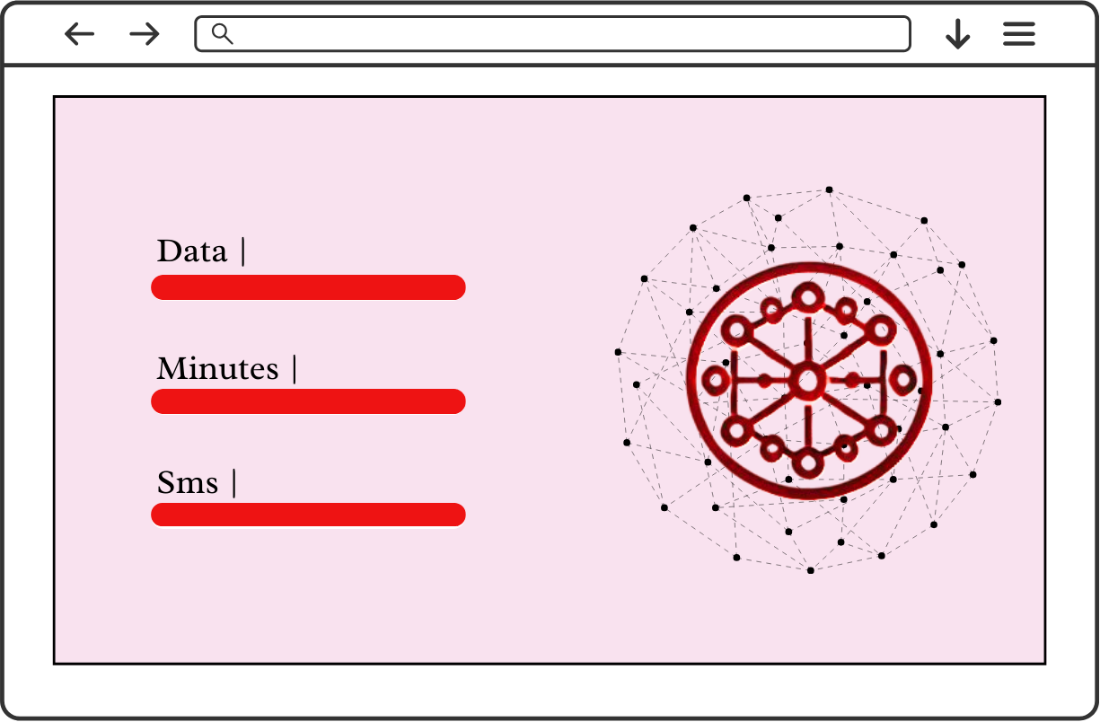
**Figure.7** Register Page

(Web)

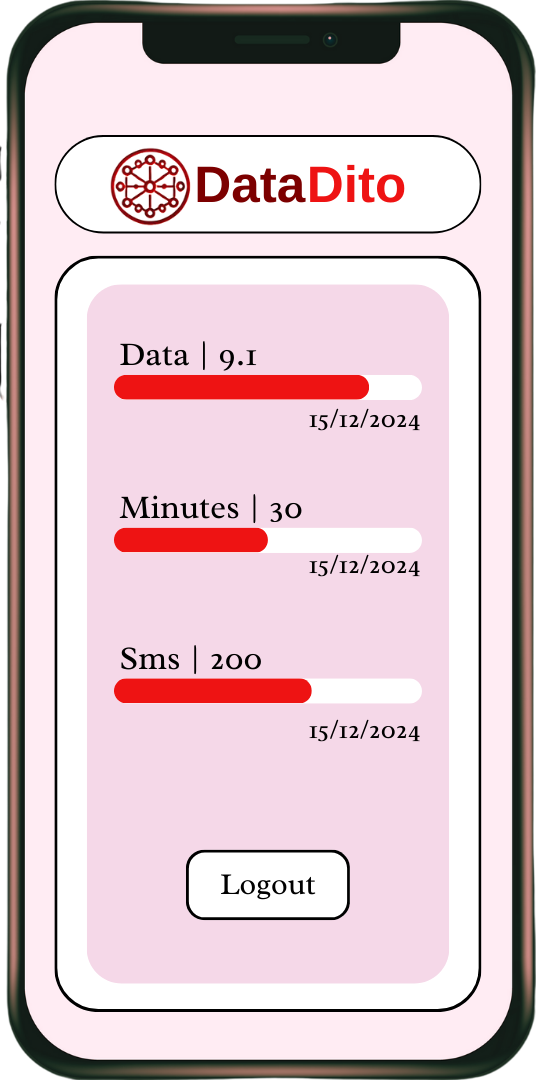


**Figure.8**

User Interface (Web)

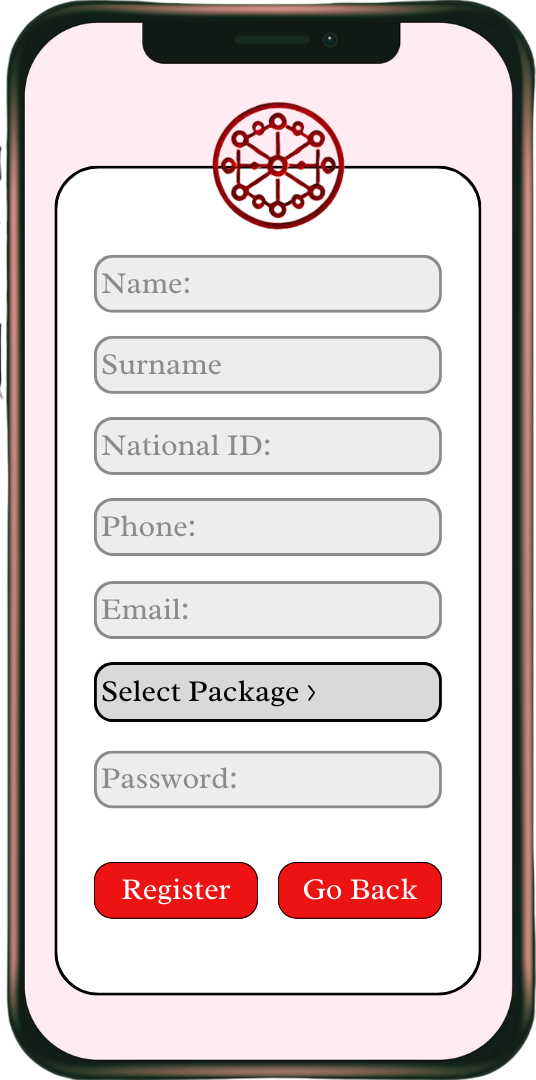


* 1. Real-time Solution Overview

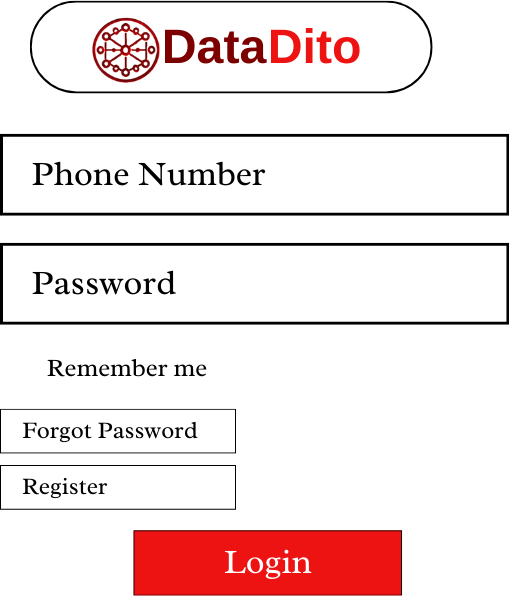
**Figure.10** Login Page (Mobile/Android)

**Figure.9** User Interface (Mobile/Android)

**Figure.11** Register Page (Mobile/Android)

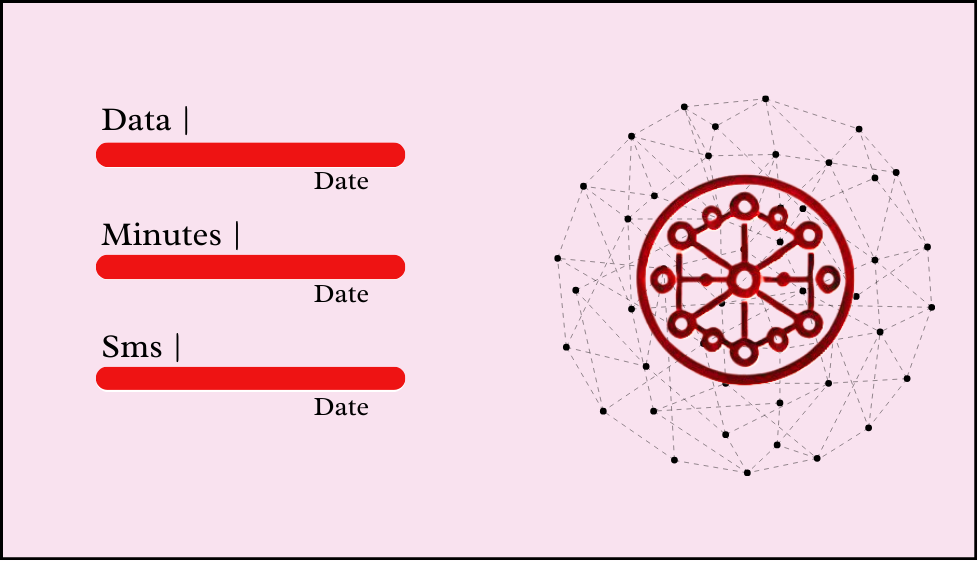
**Figure.12** User Interface (Mobile/Android)



**Figure.13**

Login Page

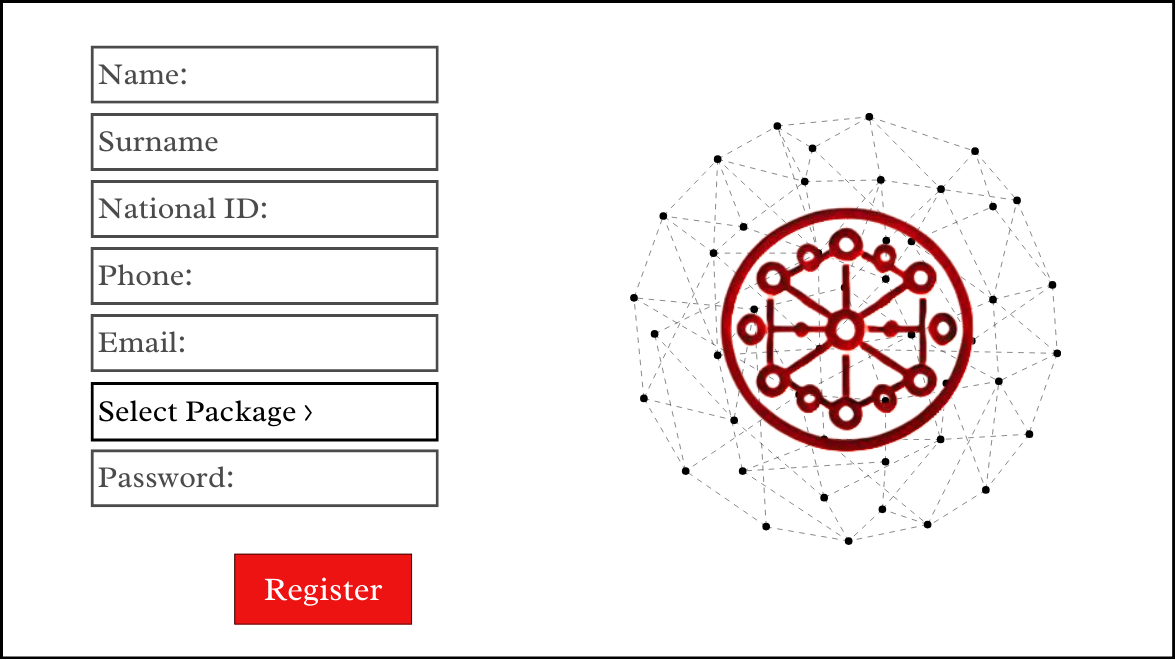
(Web)



**Figure.14**

User Interface

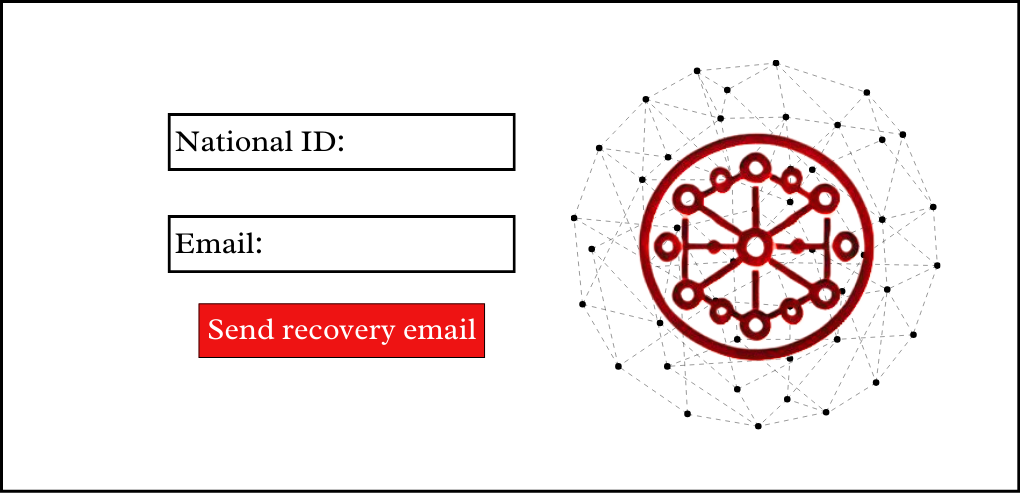
(Web)



**Figure.15**

Register Page

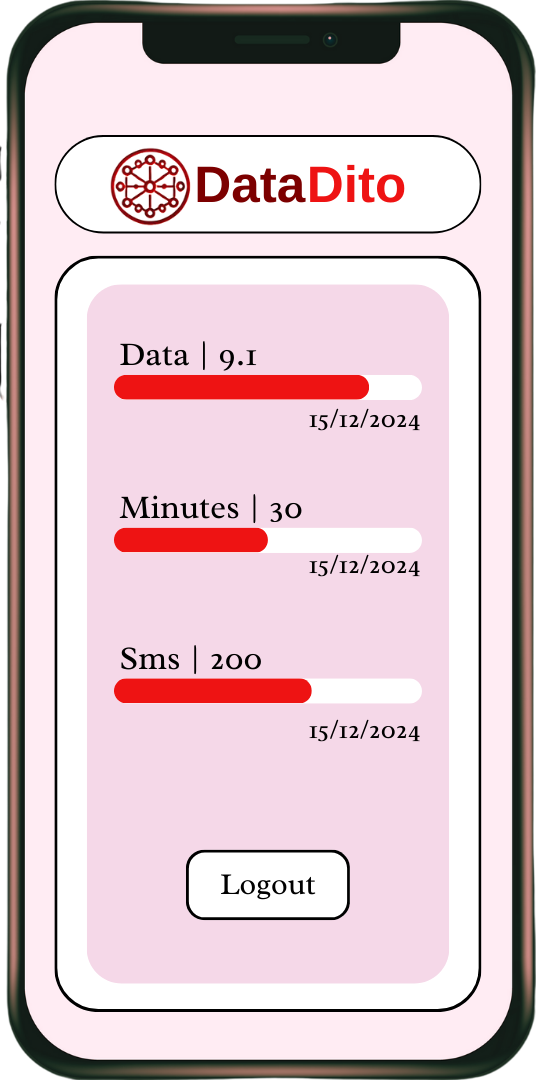
(Web)



**Figure.16**

Forgot Password Page

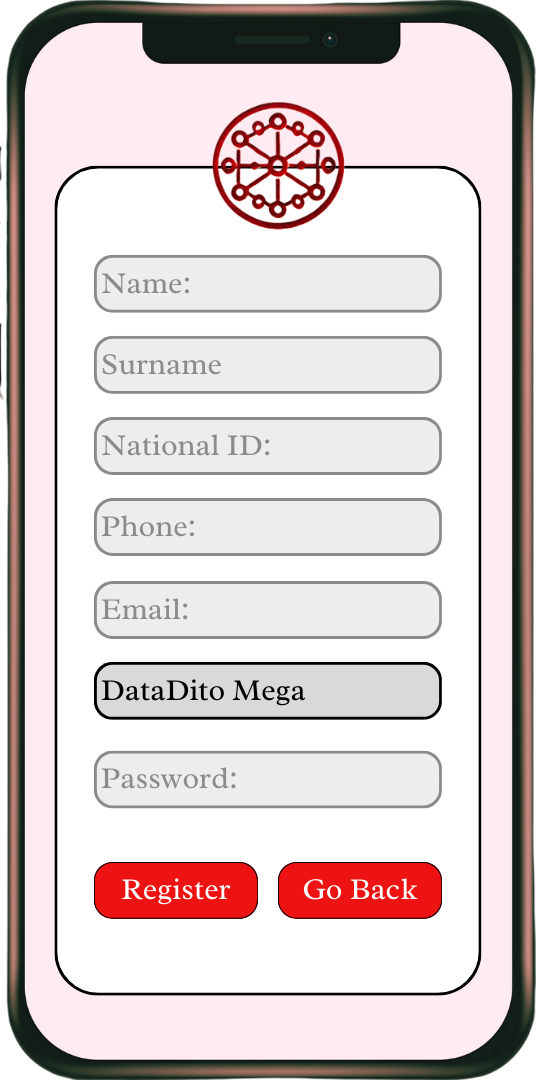
(Web)

**Figure.17** Login Page

(Mobile/IOS)

**Figure.18** User Interface (Mobile/IOS)

**Figure.19** Register Page (Mobile/IOS)

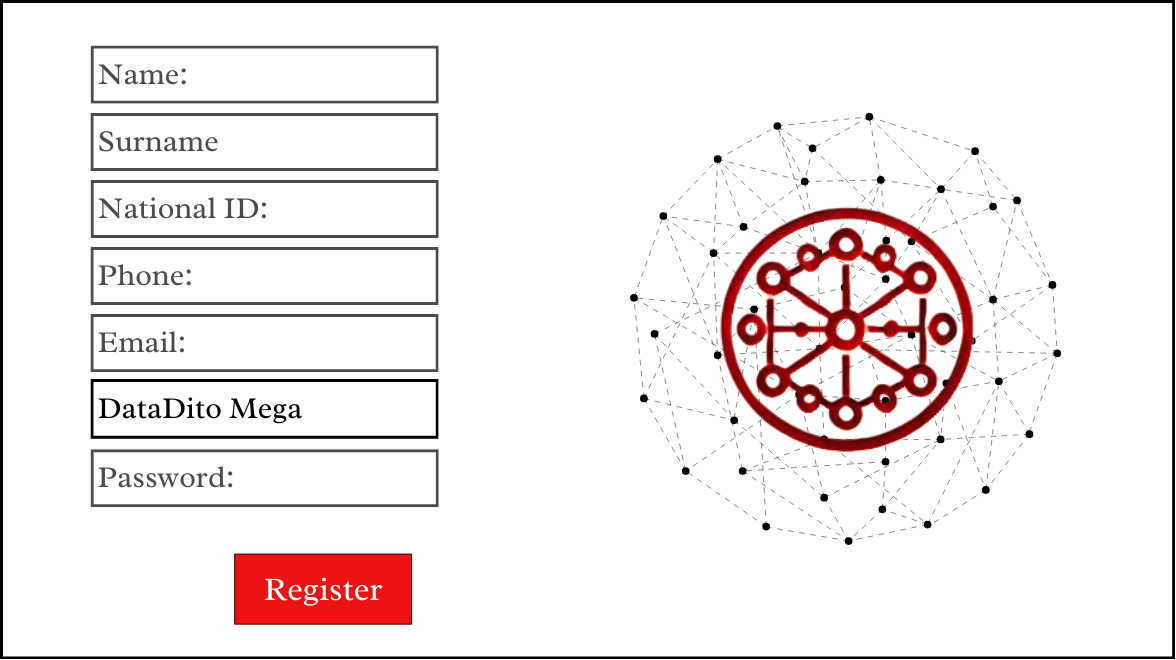
**Figure.20** Forgot Password Page

(Mobile/IOS)

## 

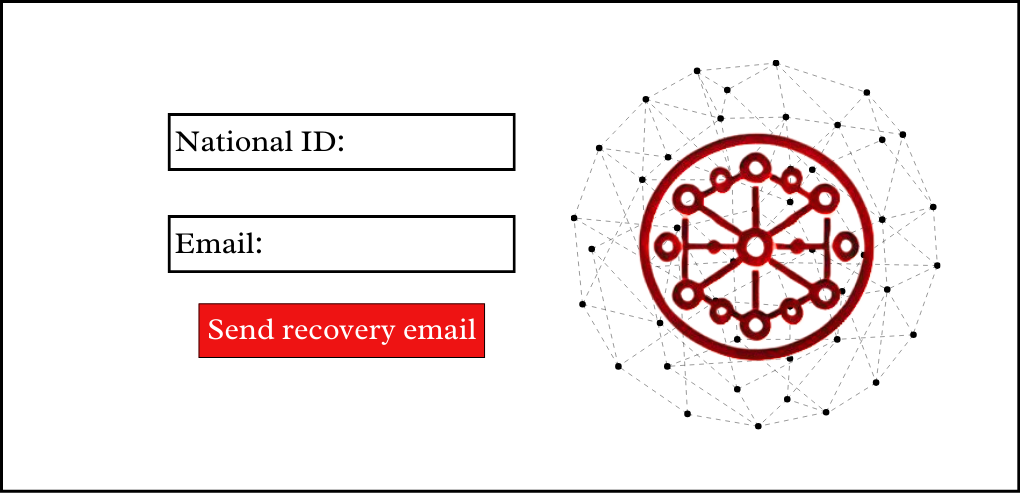
**Figure.21** Login Page

(Desktop App)



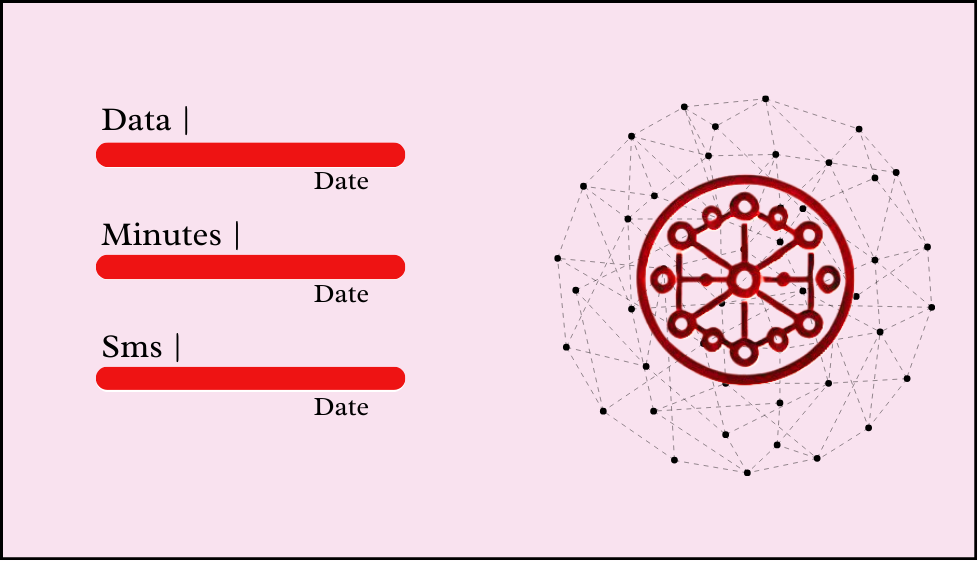
**Figure.22**  Register Page

(Desktop App)



**Figure.23** Forgot Password Page

(Desktop App)



**Figure.24** User Interface

(Desktop App)



**Figure.25**

Remaining Usage Message

(SMS)



**Figure.26**

Outcome for “Wrong Format of Message”

(SMS)

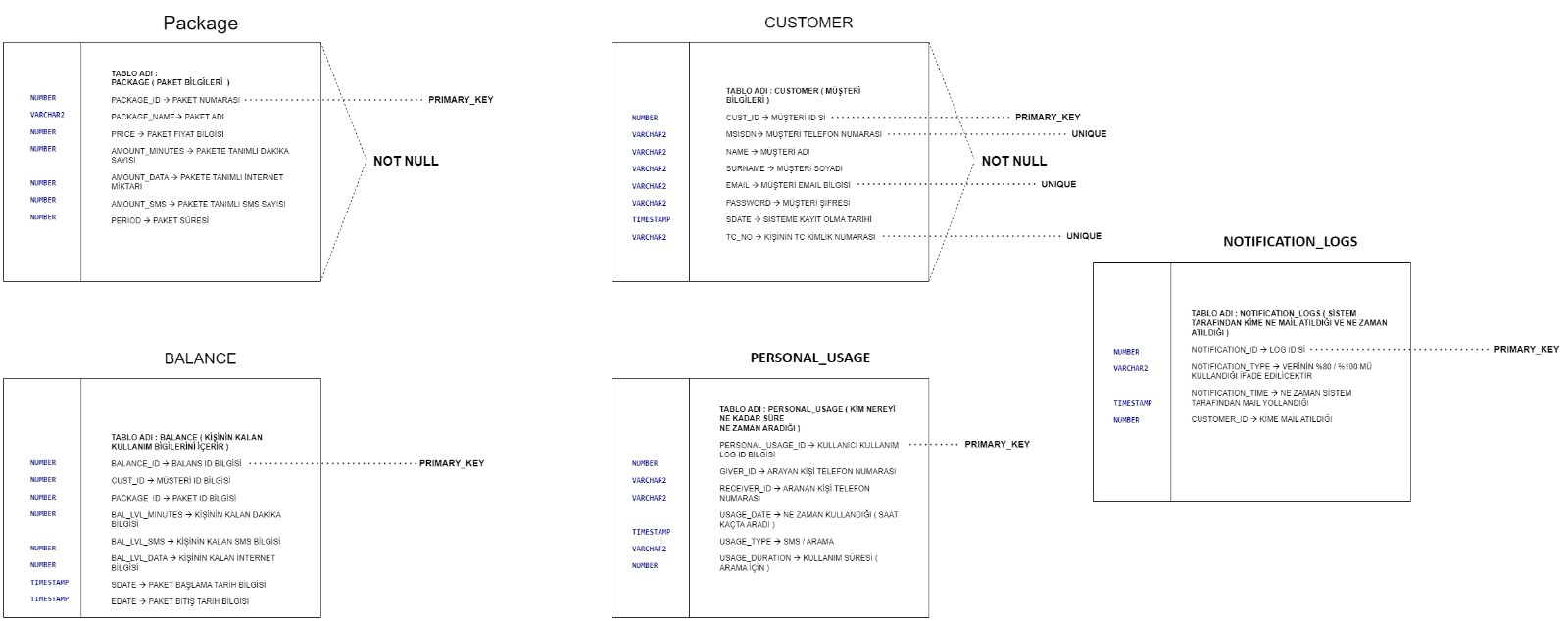


**Figure.27**

Outcome for “No Registered User for This Number”

(SMS)

## Database Solution Overview



## High-Level Design Details

metin, diyagram, ekran görüntüsü, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure.28** MessageDeliveryDiagram

metin, diyagram, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure.29** Oracle Data Synchronization

**metin, diyagram, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Figure.30** VoltDB Data Synchronization

# Diagrams

diyagram, çizgi, teknik çizim, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure.31** Flowchart (Web, Desktop, IOS, Android)

**metin, diyagram, daire, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Figure.32** Flowchart (Login/Forgot Password)

**diyagram, çizgi, taslak, daire içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Figure.33** Flowchart (Register)

# End to End User Cases

## Use Case – 1

|  |  |
| --- | --- |
| **Summary** | Successful Login |
| **Impacted System Components** | CHF, TGF, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | In order for the subscriber to log into the system successfully, users must enter their information correctly. |
| **Post-conditions** | The user continues to log in to the system with the new password. |
| **Main Success Scenario** | 1-User enters their phone number and password information on the Login Screen and clicks Login button.  2-A check is made to match the user data in the database with the  entered information. If the information is correct, the user logs into the system. |
| **Alternate Flows** | Unsuccessful Login |
| **Exceptions/Errors** | If the user enters the wrong phone number and/or password, they cannot log into the system. |
| **Open Items** | - |
| **Functional Requirements** | 6 |

## User Case – 2

|  |  |
| --- | --- |
| **Summary** | Unsuccessful Login |
| **Impacted System Components** | CHF, TGW, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | In order for the subscriber to log into the system successfully, users must enter their information correctly. |
| **Post-conditions** | The subscriber cannot log in to the system and receives a warning message. |
| **Main Success Scenario** | 1-User enters their phone number and password information on the Login Screen and clicks Login button.  2-A check is made to match the user data in the database with the entered information. If the information is incorrect, the user cannot log in to the system and receives a warning message. |

|  |  |
| --- | --- |
| **Alternate Flows** | Successful Login |
| **Exceptions/**  **Error rs** | If the information is correct, the user logs into the system. |
| **Open Items** | - |
| **Functional Requirements** | 6 |

## User Case - 3

|  |  |
| --- | --- |
| **Summary** | Forget Password |
| **Impacted System Components** | CHF, TGW, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | The user must be registered in the system. |
| **Post-conditions** | The user logs into the system with the new password. |
| **Main Success Scenario** | 1-The user clicks the "Forgot Password" button on the Login Screen.  2-The user enters their e-mail address.  3-The user enters their National ID  4-The security code will be sent to e-mail addresses of the users.  5-The user will enter the code from the e-mail and click the change password button. |

## Use Case – 4

|  |  |
| --- | --- |
| **Summary** | Change Password |
| **Impacted System Components** | CHF, TGW, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | The user must be entering security code. |
| **Post-conditions** | The user logs into the system with the new password. |
| **Main Success Scenario** | 1. The user will enter their new password. 2. The user will re-enter the new password. 3. The user should click on the change my password button. |

## Use Case – 5

|  |  |
| --- | --- |
| **Summary** | Sign Up |
| **Impacted System Components** | CHF, TGW, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | The user must have an account. |
| **Post-conditions** | User successfully registers in the system |
| **Main Success Scenario** | 1. User registers to he/she system with her/his name, surname, password, phone number, Turkish ID number or Passport number, e-mail and security key.      1. User must select one of the available packages. 2. The data registered by the user is saved in the database. |

|  |  |
| --- | --- |
| **Alternate Flows** | Unsuccessful Create Costumer |
| **Exceptions/**  **Errol rs** | 1. In case of missing or incorrect data, the subscriber will not be able to register. 2. user must select an available package. The user cannot register without selecting an existing package. |
| **Open Items** |  |
| **Functional Requirements** | 6 |

## Use Case – 6

|  |  |
| --- | --- |
| **Summary** | Successful Inquire balance |
| **Impacted System Components** | CHF, TGF, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | Subscriber must be successfully logged onto the system. |
| **Post-conditions** | Subscriber successfully sees his/her remaining Data, Minute and SMS  allowances. |
| **Main Success Scenario** | 1-Subscriber successfully logs onto the system.  2-Subscriber's remaining balances are inquired from the database.  3-Subscriber successfully sees his/her remaining Data, Minute and  SMS allowances. |
| **Alternate Flows** | - |
| **Exceptions/Errors** | - |
| **Open Items** | - |
| **Functional Requirements** | 6 |

## Use Case – 7

|  |  |
| --- | --- |
| **Summary** | Successful Inquire balance |
| **Impacted System Components** | CHF, TGF, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | Subscriber must be successfully logged onto the system. |
| **Post-conditions** | Subscriber successfully sees his/her remaining Data, Minute and SMS  allowances. |
| **Main Success Scenario** | 1-Subscriber successfully logs onto the system.  2-Subscriber's remaining balances are inquired from the database.  3-Subscriber successfully sees his/her remaining Data, Minute and  SMS allowances. |
| **Alternate Flows** | - |
| **Exceptions/Errors** | - |
| **Open Items** | - |
| **Functional Requirements** | 6 |

## Use Case – 8

|  |  |
| --- | --- |
| **Summary** | Successful Package Upgrade |
| **Impacted System Components** | CHF, TGF, AOM, HAZELCAST, VOLTDB, KAFKA |
| **Pre-conditions** | -The user must have successfully logged into the system.  - The user must have an active package. |
| **Post-conditions** | -The user's package is updated in the system.   |  | | --- | | - The user successfully upgrades to a new package. |  |  | | --- | |  | |
| **Main Success Scenario** | 1-The user clicks on the "Upgrade Package" button on the main screen.  2-The user selects a new package from the available list.  3-The user confirms the selection by clicking the "Confirm" button.  4-The system validates the user's information and processes the upgrade.  5-A success message, "Your package has been successfully upgraded," is displayed. |
| **Alternate Flows** | - |
| **Exceptions/Errors** | - |
| **Open Items** | - |
| **Functional Requirements** | 6 |

# 

# Appendix

## Document Control

### References

Online Charging Sistemi Projesi.docx

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Descriptions** | **Author** | **Date** |
| 1.0.0 | Initial Version | Zeynep Özer | 22.11.2024 |
| 1.1.0 | 2nd Version | Zeynep Özer | 24.11.2024 |
| 1.1.1 | 3rd Version | Zeynep Özer | 01.12.2024 |