|  |  |
| --- | --- |
| **Client** | i2iCell Clients |
| **Project Name** | i2iCell Android Application |
|  |  |

Table of Contents

1. Document Overview 4
   1. Scope & Purpose 4
   2. Version Numbers 4
   3. Audience 4
   4. Document Structure and Special Conventions 4
   5. Definitions, Symbols and Abbreviations 4
      1. Definitions & Terminology 4
      2. Abbreviations 5
   6. References 6
2. Overall Design Considerations 7
   1. Assumptions 7
   2. Prerequisites, Restrictions and Constraints 7
   3. Design Methodology 7
   4. Open Issues and Risks 8
3. High Level Architecture 9
   1. i2iCell Service Login 9
   2. i2Cell Service Change Password 9
   3. i2iCell Service Sign Up 9
   4. i2iCell Service Balance 9
4. Low Level Architecture 10
   1. Application, Component, or Module Name-1 11
   2. Application, Component, or Module Name-2 12
5. Interfaces & Integration 14
   1. BURAYA SCREENSHOTLAR and Integration Map 14
   2. External Interfaces Map 15
   3. System/Solution Interfaces Map 16
6. Deliverables Portfolio; Applications and Libraries 17
   1. i2i developed Applications and Libraries 18
   2. 3rd Party Applications and Libraries 21

Appendix 21

Document Control 21

# Document Overview

## Scope & Purpose

I2iCell Android Application projesi i2iCell müşterlerinin kalan bakiye bilgilerini Android akıllı telefon ortamlarında edinmelerini sağlayan bir uygulama projesidir. Elde edilen ürün, kullanıcılara hatlarına internet, sms ve dakika hakkı tanımlama ve daha önce tanımlamış oldukları paketlerden kalan kullanım bakiyelerini görme imkanı sunar.

## Version Numbers

## Audience

Bu döküman aşağıdaki kişilerce incelenebilir:

* Business/System Analysts
* Solution/System Architects

Bu döküman aşağıdaki kullanıcılara yol göstericidir:

* I2iCell Clients

## Document Structure and Special Conventions

Tasarlanan uygulamada toplamda dört ekran bulunmaktadır. Bunlar:

* Kullanıcı Girişi/ Login
* Yeni Kullanıcı Oluşturma/ Sign Up
* Parola Değiştirme/ Change Password
* Kalan Kullanım Bilgilerini Görüntüleme/ Show Balance

## Definitions, Symbols and Abbreviations

### Definitions & Terminology

|  |  |  |
| --- | --- | --- |
| **Turkish** | **Entity Name** | **Description** |
| Kullanıcı Girişi | User Login | Bu sayfa kullanıcıların daha önce uygulamada tanımlamış oldukları telefon numarası ve parola bilgilerine dayanarak giriş yapmalarını ve kalan kullanım ekranına geçebilmelerini sağlar. Ayrıca parola değiştirme ve yeni kullanıcı tanımlama ekranlarına geçişler de bu ekran üzerinden sağlanır. |
| Parola Değiştirme | Change Password | Bu sayfada kullanıcı, sisteme kayıt olurken belirttiği TC kimlik numarası ve telefon bilgileriyle yeni parola belirler. |
| Yeni Kullanıcı Tanımlama | New User Sign Up | Bu sayfada kullanıcılar sisteme ilk kayıt işlemlerini gerçekleştirirler. |
| Kalan Kullanım Görüntüleme | Show Balance | Bu sayfada kullanıcının paket tanımlama ve kalan kullanım bilgilerini gösterme etkinlikleri bulunur. |

### Abbreviations

|  |  |
| --- | --- |
| TERM | DESCRIPTION |
| API | Application Programming Interface |
| ASN.1 | Abstract Syntax Notation 1. An abstract syntax notation, which is used to describe data and data structure. |
| BGW | Billing Gateway |
| BRS | Business Requirements Specification |
| BSS | Business Support System |
| CDR | Call Data Record – Records in files generated by external network nodes such as MSC, SDP etc. |
| DWH | Data Ware House |
| EDIFACT | Electronic Data Interchange For Administration, Commerce and Transport. EDIFACT indicates a format |
| EIR | Equipment Identity Register |
| FMS | Fraud Management System |
| GUI | Graphical User Interface |
| HTML | Hyper Text Markup Language. |
| HTTP | Hyper Text Transfer Protocol. Simple stateless protocol used throughout the internet. |
| IVR | Interactive Voice Response System |
| Java | An object orientated programming language which is portable between platforms. |
| MML | Man Machine Language, specifies the command format API of the HLR |
| OSS | Operations Support System |
| SAS | System Architectural Specification |
| SOAP | Simple Object Access Protocol  SOAP is a simple XML-based protocol to let applications exchange information over HTTP |
| SRS | System Solution Requirements Specification |
| VAS | Value Added Service |
| XML | Extensible Markup Language. A widely used standard mark-up language. |
| XSD | XML Schema Definition. XML Schema is an XML based alternative to DTD. An XML schema describes the structure of an XML document. |

## References

1. SRS\_SystemRequirementsSpecifications\_[Area]\_[ProjectCode]\_v1.1.5.doc, System Requirements Specifications, Jan 2014, i2i-Systems
2. NFS\_NonFunctionalSpecifications\_[Area]\_[ProjectCode]\_v1.1.4.doc, Use Cases Specification, Jan 2014, i2i-Systems

# Overall Design Considerations

## Assumptions

Describe any assumption, background, or dependencies of the system/solution, its use, the operational environment, or significant project issues that impact overall design

## Prerequisites, Restrictions and Constraints

Describe any constraints on the system that have a significant impact on the overall design of the system. (e.g. technology constraints, performance requirements, end user characteristics, validation requirements, project constraints, etc.)

1. Abc
2. Abc

## Design Methodology

Bu proje oluşturulurken Agile Tasarım Metodolojisi benimsenmiştir.

## Open Issues and Risks

*Proje herhangi bir risk ihtiva etmez.*

# High Level Architecture

String myURL ="http://68.183.75.84:8080/i2iCellService/services/Services/" +  
 "login?inputPhoneNumber="+phoneNumber.getText()+"&inputPassword="+loginPassword.getText();  
URL url = new URL(myURL);  
HttpURLConnection conn=(HttpURLConnection) url.openConnection();  
conn.setConnectTimeout(60000);  
BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));  
String str;  
while ((str = in.readLine()) != null) {  
 Log.*i*("",str);  
 response += str;  
}

Figure 1: High-Level Architecture

## i2iCell Service Login

<http://68.183.75.84:8080/i2iCellService/services/login>

Bu servise telefon numarası ve parola olmak üzere iki veri gönderilir. Servis response’u 1 vey 0 şeklinde döner. 1, veritabanı ile işleşen kullanıcı girişi olduğunu, 0 ise verilen bilgilerin veritabanına kaydedilmiş verilerle eşleşmediği bilgisini içerir. Bu bilgi, uygulamanın backend kısmında anlamlı hale getirilir.

## i2Cell Service Change Password

<http://68.183.75.84:8080/i2iCellService/services/changePassword>

Bu servise telefon numarası, TC kimlik numarası ve parola bilgileri gönderilir. Webservice response olarak 0 veya 1 karakterlerini post eder. Bu veriler bir String değişkeni içerisine alınarak, bilgilerin doğruluğu veya yanlışlığına göre anlamlı hale getirilerek kullanıcıya dönüt verilir.

## i2iCell Service Sign Up/Create Account

<http://68.183.75.84:8080/i2iCellService/services/createAccount>

Bu serviste yeni kullanıcılar TC kimlik numarası, telefon numarası, isim, soyisim, e-mail, doğum tarihi ve şifre bilgilerini girer. Webservice, bu bilgilerle veritabanındaki user\_info tablosuna insert işlemi gerçekleştirir.

## i2iCell Service Balance

<http://68.183.75.84:8080/i2iCellService/services/getBalances>

Bu serviste login ekranından doğru bilgilerle yönlendirilmiş olan kullanıcıya kalan kullanım bilgileri gösterilir.

# Low Level Architecture

String[] xmls = response.split("<ns:return>");  
String withoutReturnXML = xmls[1] + xmls[2] + xmls[3];  
String[] lastXMLS = withoutReturnXML.split("</ns:return>");  
String LasResponse = lastXMLS[0] + " " + lastXMLS[1] + " " + lastXMLS[2];  
Toast.*makeText*(getApplicationContext(), LasResponse, Toast.*LENGTH\_SHORT*).show();  
String result = "" + response.charAt(62);  
gb.setText(lastXMLS[0]);  
dk.setText(lastXMLS[1]);  
sms.setText(lastXMLS[2]);

Figure 2: Low-Level Architecture

## Application, Component, or Module Name-1

|  |  |
| --- | --- |
| ***Module Properties*** | ***Property Details*** |
| ***Name*** | APRMWEB |
| ***Module Category***  ***(Application / Library / Other )*** | Application |
| ***Module Type***  ***(PLSQL / EJB / WebService / Standalone / Library/Web Server)*** | Web Server |
| ***Shipment Type***  ***(JAR / EAR / WAR / BINARY EXE / SCRIPT / (UN)WRAPPED PLSQL)*** | EAR |
| ***Interaction Type***  ***(Synchronous/Asynchronous)*** | Asynchronous (with web browser on client side)  Synchronous with APRMWS (APRM Web Service) |
| ***DB Dependency*** | NO |
| ***Expected Latency*** | ~50ms |
| ***Access Level***  ***(Public / Non-Public / 3rd Party)*** | Internal |
| ***Input*** | From ZK Update Engine information submitted from relevant web page |
| ***Output*** | TXDR |
| ***Main Processing***  ***Logic*** | APRMWEB is the Model/View and Controller layer developed compliant with ZK framework.  Views are provided as ZUL files.  Model Layer is lightweight and no business logic related model objects are managed in this layer.  Controller is a relay in the sense that each page requests the relevant Back-End Service operation by invoking Execute(TXDR) web service.  TXDR contains all information with regards to requested operation and the payload for that operation.  This provides a message centric approach, and web service layer is only used as a transport layer and does NOT need to be modified for different types of business operations, making web service layer intact from business logic. |

The purpose of this section is to highlight the specific role, responsibility and the main functionality fulfilled by the relevant Application Module within the overall architecture.

*Sample:*

## Application, Component, or Module Name-2

The purpose of this section is to highlight the specific role, responsibility and the main functionality fulfilled by the relevant Application Module within the overall architecture.

*Sample:*

|  |  |
| --- | --- |
| ***Module Properties*** | ***Property Details*** |
| ***Name*** | REQIF |
| ***Module Category***  ***(Application / Library / Other )*** | Application |
| ***Module Type***  ***(PLSQL / EJB / WebService / Standalone / Library)*** | Standalone |
| ***Shipment Type***  ***(JAR / EAR / WAR / BINARY EXE / SCRIPT / (UN)WRAPPED PLSQL)*** | JAR |
| ***Interaction Type***  ***(Synchronous/Asynchronous)*** | Synchronous |
| ***DB Dependency*** | NO |
| ***Expected Latency*** | ~20ms |
| ***Access Level***  ***(Public / Non-Public / 3rd Party)*** | Public |
| ***Input*** | GDR |
| ***Output*** | GDR |
| ***Main Processing***  ***Logic*** | REQIF is an internal high-throughput (non-public) common low-level Synchronous to Asynchronous Technical GW of APRM.  REQIF decouples the external synchronous web service layer, from the high throughput asynchronous processing sub-system (TMF).  Processing logic is as follows;   * Get GDR and block synchronous RMI REQIF.execute(gdr) call * Generate INTL\_TXN\_ID * Register request to REQUEST\_INPUT and REQUEST\_INPUT\_DETAIL. Set REQUES\_INPUT.TXN\_STATUS=’R’ indicating request registered. * MXLIB.WRITE (GDR) to JMS TMF-Q * MXLIB.commit & DB.COMMIT * Wait for reply in JMS: MXLIB.READ from REQIF-Q * Update TXN\_REPLIED 'Y' on REQUEST\_INPUT * Find blocked RMI call for the relevant INTL\_TXN\_ID * Return back to RMI client * MXLIB.COMMIT (to DELETE the READ reply message from JMS-Q) * DB.COMMIT |

# Interfaces & Integration

**Installation Guide**

* Open README file of the project on GitHub page “<https://github.com/i2icell/Android_AysegulKarahancer/>”
* Follow the instructions.
* All steps are explained with their specific screenshots.

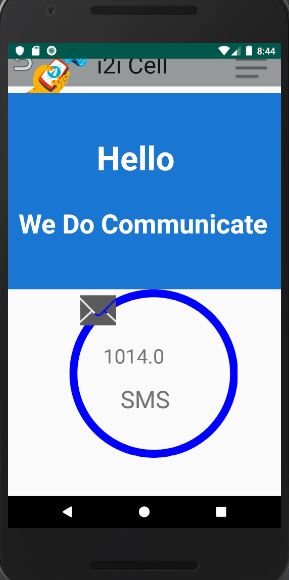
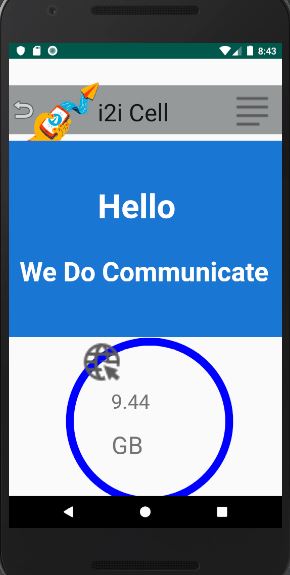
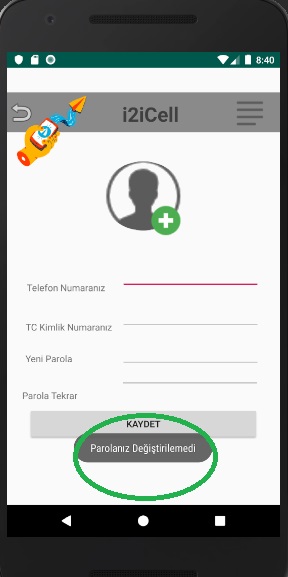
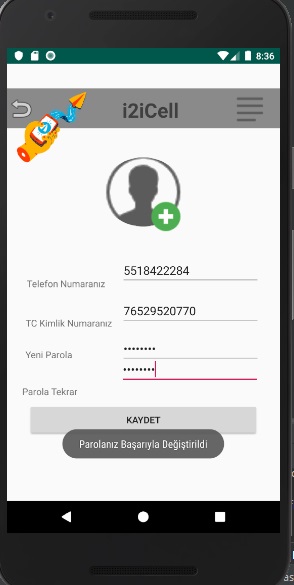
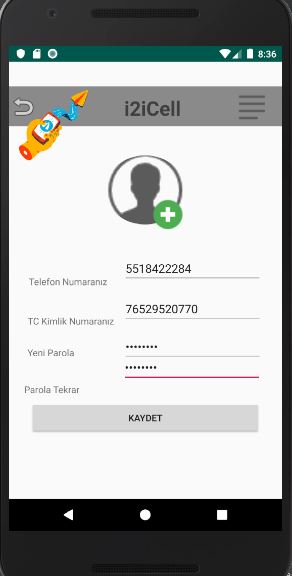
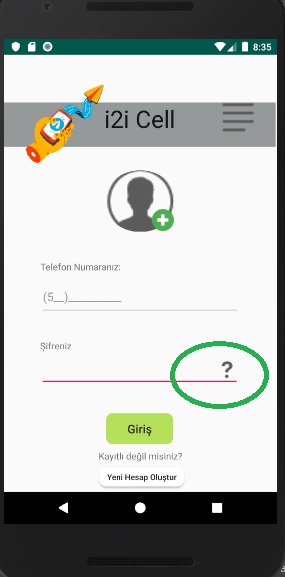
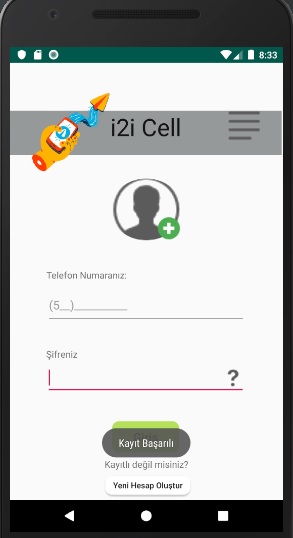
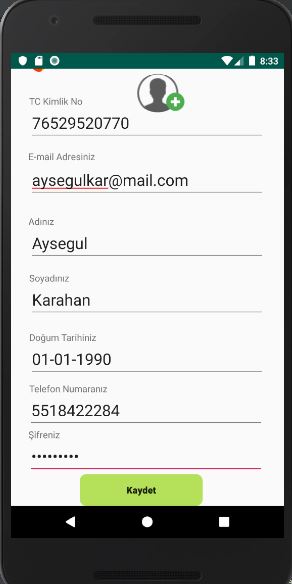
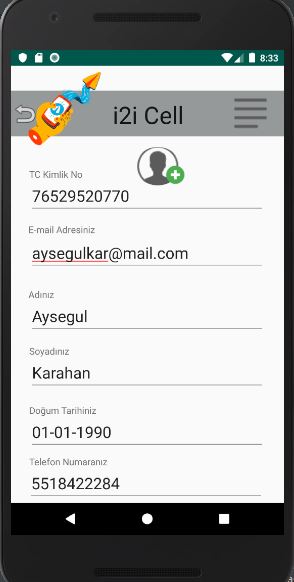
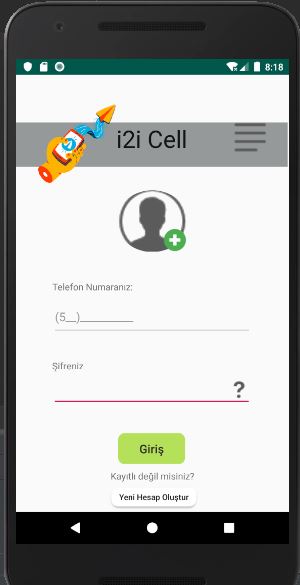


Figure 3: Interfaces and Integration Map

## External Interfaces Map

*This section provides a list of the interfaces provided by the systems outside the scope of the solution/system of our concern. In other words this section lists the interfaces that are external. For the actual specification of interfaces, please refer to relevant interface specification documents.*

## System/Solution Interfaces Map

*This section provides a list of the publicly accessible*

*and private interfaces provided by the system/solution. Note that since private interfaces are used internal and not supported, only the critical ones can be provided as additional information. For the actual specification of interfaces, please refer to relevant interface specification documents.*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Source** **Applications/**  **Modules** | **Destination** **Applications/**  **Modules** | **Protocol** | **Client**  **Interaction Type** | **Frequency/**  **Response** | **Type** | **Interface**  **Packaging** | **Access**  **Level** |
| 1 | … | … | … | Web Service http-soap | Synchronous | <5sec | Online | Java APIs | Public |
| 2 | … | … | … | Web Service http-soap | Synchronous | <5sec | Online | Java APIs | Public |
| 3 | … | … | … | File Based  ASN.1 | Asynchronous, | 10 minutes | scheduled batch | N/A | Public |
| 4 |  |  |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |  |
| 7 | … | … | … | ASCII over TCP | Synchronous | <300ms | Realtime | C shared library | Private |

# Deliverables Portfolio; Applications and Libraries

*This section provides a list of deliverables from deployment perspective where a list of all i2i developed and/or 3rd party application and libraries is to be provided.*

## i2i developed Applications and Libraries

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Module Category**  **(Application/**  **Library)** | **Shipment Type** |
| **APRMWEB** | View Layer and Basic Controller. Set of ZUL files and basic controller related Java classes on View Layer. | APPLICATION | EAR |
| **APRMWS** | APRM Web Service | APPLICATION | EAR |
| **REQIF** | Request Interface RMI Gateway | APPLICATION | JAR |
| **TMF** | Transaction Mgmt Function | APPLICATION | JAR |
| **INT-E** | Integration Engine | APPLICATION | JAR |
| **BACH** | Batch Operations Handler | APPLICATION | JAR |
| **GDR.xsd**  **GDR.jar** | Generic Data Record XML Schema Specification | LIBRARY | JAR |
| **TXDR.xsd**  **TXDR.jar** | Transaction Data Record XML Schema Specification | LIBRARY | JAR |
| **ENTITY.xsd**  **ENTITY.jar** | Entity XML Schema Specification that holds all PRM related entity object model; Dealer, Position, User etc. used for transport of all business value objects over Web Service, RMI and JMS layers. | LIBRARY | JAR |
| **CHARSPEC.xsd**  **CHARSPEC.jar** | Screen Objects and characteristics XML Schema Specification | LIBRARY | JAR |
| **I2iCommon.xsd**  **I2iCommon.jar** | Common Value Object types | LIBRARY | JAR |
| **WORKFLOW.xsd**  **WORKFLOW.jar** | Workflow Model XML Schema Specification | LIBRARY | JAR |
| **WFELib**  **(wfecore.jar and**  **wfeTMF.jar)** | Workflow Library providing workflow API | LIBRARY | JAR |
| **BIZLib.jar** | Business Library that provides all DB layer related update logic | LIBRARY | JAR |
| **EFELib.jar** | Enterprise Function Evaluator Library that provides a configurable action processing framework used by INT-E and WFELib | LIBRARY | JAR |
| **INT\_DSF.jar** | DSF Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_INFODEALER.jar** | INFODEALER Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_OM.jar** | OM Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_BSCS.jar** | BSCS Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_MERNIS.jar** | MERNIS Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_SAP.jar** | SAP Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_LDAP.jar** | LDAP Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **INT\_LDAP\_DEALER.jar** | Dealer LDAP Integration Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **MAIL.jar** | MAIL sending Plug-In Functions API functions that implement EFE interfaces | LIBRARY | JAR |
| **WFE\_FUNC.jar** | Other WFE related Plug-In API functions that implement EFE interfaces | LIBRARY | JAR |
| **dtflib.jar**  **DecisionTreeSchema.jar**  **DecisionTreeSpecification.jar** | Decision Tree Library that allows configurable message routing and determination of workflow transition and selection rules | LIBRARY | JAR |
| **Mxlib.jar** | Message Exchange Library for configurable inter-application message routing | LIBRARY | JAR |
| **ithreadpool.jar** | Configurable multi-threaded and reliable application framework | LIBRARY | JAR |

## 3rd Party Applications and Libraries

*The below table depicts the most critical 3rd party libraries used by the System/Solution. Since versions may change at time of deployment, they are not provided.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Module Category**  **(Application/**  **Library)** | **Shipment Type** |
| **Axis.jar** | Apache Axis related to web services | LIBRARY | JAR |
| **Commons-xxx.jar** | Apache commons libraries;   * Commons-logging * Commons-cli * Commons-dbcp * Commons-dbutils * Commons-lang * Commons-pool * Commons-beanutils | LIBRARY | JAR |
| **Junit.jar** | JUnits related JAR | LIBRARY | JAR |
| **Log4j.jar** | Logging related JAR | LIBRARY | JAR |
| **Saxon9he.jar**  **Saxpath.jar** | XML processing related JAR | LIBRARY | JAR |
| **Xbean.jar**  **Xbean.xpath.jar** | XML Beans related JAR | LIBRARY | JAR |
| **Pop3.jar**  **Smtp.jar**  **Mailapi.jar** | Mail related JAR | LIBRARY | JAR |
| **Velocity.jar** | Template driven development JAR  Used for email templates and other integration purposes | LIBRARY | JAR |
| **Telnetd.jar**  **Dsn.jar** | TCP/IP based communication JAR | LIBRARY | JAR |

# Appendix

# Document Control

Superseded Documents

* N/A

Change Control & Distribution

|  |  |
| --- | --- |
| **Owner** | [OWNER NAME] |
| **Reviewer** |  |
| **Approved By** |  |
| **Distribution** |  |
| **File Name** | SAS\_SystemArchitectureSpecifications\_[ProjectCode]\_Template\_v1.1.3 |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Descriptions** | **Author** | **Date** |
| 1.0.0 | Initial Version | [Author Name] | 01/Jan/2014 |
|  |  |  |  |

Approvals

This document requires the following approvals:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Title** | **Date** | **Signature** | **Version** |
|  |  |  |  |  |
|  |  |  |  |  |