i2i Systems
innovation to integration

Test Cases

Client	[All digital operators]
Project Name	[ech0cell]

Table of Contents

1 Document Overview

1.1 Scope & Purpose

The scope of this document is to provide i2i Internal Acceptance Test Cases and their results on [Android app] area which forms a critical part of the [ech0cell] Program/Project of [Android testing].

1.2 Version Numbers

This Specification has been produced by the i2i Systems and Consultancy (i2i) following 3GPP specifications alike versioning scheme.

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.

1.3 Audience

The document may be read by the below audience;

- Business departments
- Business/System analysts
- Solution/System Architects
- Program/Project Managers

1.4 Document Structure and Special Conventions

If the document is complex and specific sections apply to specific reader profile, a brief description of vital sections contained in the document shall be provided.
Such as;

Section X: Explains the details related to xyz providing diagrams. This section shall be read by Business Analysts
Section Y: Explains the details related to wqv from system perspective. This section shall be read by Solution Architects
Etc.

1.5 Definitions, Symbols and 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.

1.6 Reference Documents

[1]   Reference 1
[2]   Reference 2

2 Test Execution

2.1 Test Case Index

2.1.1 TC_001: User Registration		

Short Description: Verify user registration with valid details

Scenario Description: User attempts to register using valid information.

Pre-Condition: User is not registered.

Input Data:

1)Name: "Test User"

2)Email: "testuser@example.com"

3)Password: "ValidPassword123"

Triggering the Test Case:

1) Open the app and navigate to the registration screen.

2) Enter valid user details.

3) Click on the "Register" button.

Output/Result Data:

Screenshot of successful registration message.

Expected Outputs (Success conditions)	Passed	Finding
1) User is successfully registered. 2) User is navigated to the login screen. 3) Confirmation email is received.	**YES**	

2.1.2 TC_002: User Registration with Invalid Email	

Short Description: Verify user registration with an invalid email format

Scenario Description: User attempts to register with an invalid email.

Pre-Condition: User is not registered.

Input Data:

Name: "Test User"

Email: "invalidemail"

Password: "InvalidPassword123"

Triggering the Test Case:

1) Open the app and navigate to the registration screen.
2) Enter invalid email format.
3) Click on the "Register" button.

Output/Result Data:

error message indicating invalid email.

Expected Outputs (Success conditions)	Passed	Finding
Error message is displayed indicating invalid email format.	**YES**	

2.1.3 TC_003: User Login with Valid Credentials
Short Description: Verify user login with valid credentials.
Scenario Description: A User attempts to login using valid credentials.
Pre-Condition: User is registered.
Input Data: Email: "testuser@example.com" Password: "ValidPassword123"
Triggering the Test Case: 1)Open the app and navigate to the login screen. 2)Enter valid credentials. 3)Click on the "Login" button.
Output/Result Data: Screenshot of the account details screen.

Expected Outputs (Success conditions)	Passed	Finding
User is successfully logged in. User is navigated to the account details screen.	**YES**	

2.1.4 TC_004: User Login with Invalid Password
Short Description: Verify user login with an invalid password.
Scenario Description: User attempts to login using an invalid password.
Pre-Condition: User is registered.
Input Data: Email: "testuser@example.com" Password: "InvalidPassword"
Triggering the Test Case: Open the app and navigate to the login screen. Enter valid email and invalid password. Click on the "Login" button.
Output/Result Data: Screenshot of error message indicating incorrect password.

Expected Outputs (Success conditions)	Passed	Finding
Error message is displayed indicating incorrect password.	YES	

2.1.5 TC_005: View Account Details		
Short Description: Verify user can view account details.		
Scenario Description: User logs in and views their account details.		
Pre-Condition: User is logged in..		
Input Data: N/A		
Triggering the Test Case: Open the app and navigate to the account details screen.		
Output/Result Data: Screenshot of account details view.		
Expected Outputs (Success conditions)	**Passed**	**Finding**
Account details are displayed correctly. No change of details. No crashes.	**YES**	

2.1.6	TC_006: Update Account Information

Short Description: Verify user can update account information.

Scenario Description: User attempts to update their account information..

Pre-Condition: Verify that User is logged in.

Input Data:

New Password: "NewValidPassword123"

Triggering the Test Case:

1)Open the app and navigate to the account management screen.

2)Update account information (e.g., change password).

3)Save the changes.

Output/Result Data:

Account information is updated successfully.

Confirmation message is displayed.

Expected Outputs (Success conditions)	Passed	Finding
The system should warn the user with potential missing documents and shows a message that indicates the new dealercode (SAP code) and successful activation request. Check SALES_PARTNER_* tables for this new Sales Partner and verify that correct records have been populated. Check TRANSACTIONS table and query the row with operation_type= "Sales PartnerActivation".	**YES**	

2.1.7 TC_007: Real-time Account Usage Tracking
Short Description: Verify real-time account usage tracking and balance deduction.
Scenario Description: User performs activities that use account balance and checks for real-time updates.
Pre-Condition: Verify that all SPM applications are up and running. Ensure that notification and approval e-mail account groups have been already defined in the system.
Input Data: User is logged in.
Triggering the Test Case: Activity: Making a call
Output/Result Data: Perform activities that use account balance. Check the account balance and usage in the app.

Expected Outputs (Success conditions)	Passed	Finding
Screenshot of updated balance and usage.	**YES**	

3   Test Results

3.1   Findings History

Defect ID (Mantis ID)	Short Defect Description	Opened Date	Status	Severity	Developer/ Tester
1234567		12.01.2013	Open	Critical	

4 Appendix - 1

4.1 Defect Severity Level Definitions

The severity should describe how critical the problem is. This should not describe how difficult it is to fix the problem.
Generally, there are four level of severity: Critical, High, Medium and Low.

Critical: This defect can be fatal for the execution of the system and can cause loss of business functionality and/or money for the customer. For example, if a user interface that is used to activate contracts is not working due to a software error, then this means loss of business for the customer, and the defect should be fixed urgently. Or, An SQL Error gets severity "High", because it's a fatal for the execution. But it might be easy to fix it by correcting the statement.

High/Medium: If a defect doesn't cause loss of business functionality and/or money for the customer, then this can be considered as "High" or "Medium". For example, a user interface that is used to activate contracts might populate some wrong data on database, but, the activations can be done properly despite this wrong data. The defect can be corrected within a reasonable time period. In the mean time, wrong data can be corrected with manual operations.

Low: Mostly, cosmetic defects are considered as "Low". These defects don't cause any loss of business functionality, money or data. The system can work properly, even if the defect is not fixed forever.

Appendix - 2

Document Control

Superseded Documents

- **N/A**

Change Control & Distribution

Owner	[OWNER NAME]
Reviewer	
Approved By	
Distribution	
File Name	TEST_TestCases_[Area]_[ProjectCode]_Template_v1.1.4

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