Integration Test Plan

for

VODKA

Version 1.0 Draft
April 18, 2007

Prepared by:
Archit Baweja, Drew Hall, Sunny Huynh,
Kevin Lynch, and Kanwarpreet Sethi

Drexel University
## Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason for Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archit Baweja, Drew Hall, Sunny Huynh, Kevin Lynch, and Kanwarpreet Sethi</td>
<td>1 Mar 2007</td>
<td>Initial Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>
## Contents

1 Introduction 1  
  1.1 Purpose ................................................................. 1  
  1.2 Scope .................................................................. 1  
  1.3 Definitions, Acronyms, and Abbreviations ..................... 1  

2 Approach 3  
  2.1 Objectives ................................................................. 3  
  2.2 Structure .................................................................. 3  

3 Assumptions and Exclusions 3  
  3.1 Assumptions ................................................................. 3  
  3.2 Exclusions .................................................................. 3  

4 Entry and Exit Criteria 3  
  4.1 Entry Criteria ............................................................... 3  
  4.2 Exit Criteria .................................................................. 4  

5 Participants 4  
  5.1 Roles and Responsibilities ............................................. 4  
  5.2 Reporting .................................................................. 4
1 Introduction

1.1 Purpose

This document specifies the plan for performing integration test of the VODKA financial management system, version 1.0. VODKA is a tool to manage the financial accounts of student organizations within a school or university. There are two primary goals of the VODKA system. The first goal is to allow students to manage the financial accounts of their student organizations in a simple and reliable manner. And the second goal is to provide a centralized tool by which the parent school may audit the student organizations. In order to provide these two goals, the system focuses on simplicity, flexibility, security, accessibility, and scalability. Refer to the VODKA Software Requirements Specification [2] document for additional details.

1.2 Scope

This document describes the tests involved in the integration test and the roles of test team members. Its intended audience are the designers, developers, and testers of the software system.

1.3 Definitions, Acronyms, and Abbreviations

Audit An audit is an assessment of the accuracy by which a student organization’s financial statements are presented.

Check Request A check request is a request for the parent school or university to issue a check for a specified amount to a specified recipient.

Communications Protocol A communications protocol defines the rules for sending data in a network. Each protocol may include information such as data representation, error detection, authentication and other information.

Deactivation Deactivation of a user account in the VODKA system may mean one of two scenarios. Deactivation includes deletion and disabling of user accounts. Deleting of a user account is only possible if there is no prior history associated with the account. In the case of a user account having prior history, only disabling is possible.

Finance Manager The Finance Manager is a user whose user account has privileges to create, to edit, to delete, and to view all comments, financial accounts, transactions, and user accounts of every organization within the VODKA system.

Financial Account Each financial account is an independent collection of transactions and their histories. A financial account is uniquely identified with an associated account number.

HyperText Markup Language (HTML) The HyperText Markup Language is the predominant language used for the creation of web pages.

HyperText Transfer Protocol (HTTP) The HyperText Transfer Protocol is the de facto standard protocol communications protocol for World Wide Web applications.

Internet The Internet is a publicly accessible network of interconnected computer networks.

Internet Protocol (IP) The Internet Protocol is the standard communications protocol to distinguish computers connected over a network and to send data across the Internet.

Network A computer network is the interconnection of multiple computers using a telecommunications system, which allows for the communication and sharing of resources.

Non-Volatile Storage Non-volatile storage is the type of computer memory does not lose its stored information when powered off.
Organization Advisor  An Organization Advisor is a special user account provided for each organization with the privileges to create, to edit, to delete, and to view all comments, transactions, and user accounts of the particular organization to which the advisor belongs. Each organization must have at least, but not limited to, one advisor account.

Privilege  The VODKAsoftware system limits user accounts from performing certain actions; whether a user account can or cannot perform these actions is called the user’s privileges. Privileges are to create, edit, delete, and view; each action the user can take upon the system must be classified as one of these categories. User privileges are specific to different parts of the software system.

Service-Oriented Architecture (SOA)  A software architecture that uses software services as its main components.

Simple Mail Transfer Protocol (SMTP)  The Simple Mail Transfer Protocol is the de facto standard communications protocol for email transmission over a network.

Student Organization Officer Training (SOOT)  At Drexel University, officers of student organizations must attend annual training sessions, called Student Organization Officer Training, that explain the policies and procedures involved in running student organizations. These policies and procedures include those regarding the management of financial accounts owned by the student organizations.

System Administrator  The system administrator is a special user whose user account has all the possible privileges available to any other user account as well as direct access to the underlying VODKAsoftware processes and server(s). This person(s) should be the one(s) permitted to access the physical machines running the VODKAsoftware system. The role of the system administrator is to perform system maintenance and configuration.

Transaction  A transaction is any entry into a financial account that documents financial accounting practices (e.g. recording a bank account deposit or pending check request).

Transmission Control Protocol (TCP)  The Transmission Control Protocol is a very commonly used communications protocol that guarantees transmission integrity over a network.

Transport Layer Security (TLS)  The Transport Layer Security is a set of communication encryption protocols that guarantee privacy and data integrity over a network.

User  A user is a person who interacts with the VODKAsoftware system.

User Account  A user account is an independent collection of user information that the VODKAsoftware system uses to recognize individual users. Each user account is uniquely identified by a user name and allows the software to track individual user privileges.

Volatile Storage  Volatile storage is a type of computer memory that requires power to retain the stored information. When volatile storage is powered off, all the stored information is lost.

Web Browser  A web browser is a software that allows users to view and interact with graphical and textual web sites on the World Wide Web.

2 Approach

This section describes the testing approach, tools, and technology used for integration testing of VODKA.

2.1 Objectives

The integration test process ensures the modules of the system interact with each other correctly and stably. Modules in this context refer to both classes and architectural components described in the VODKA Software Design Description [1] document.

2.2 Structure

The software is architected into three distinct layers: presentation, business, and data. Layers can only depend on those layers below it, hence the business layer can only depend on the data layer and not the presentation layer. From this, we will take an iterative “bottom-up” approach. We will start with the components of the data layer and test their integration. Afterward, we will test the integration of the business layer components. Finally, we will test the integration of the presentation layer components.

Each module will undergo integration test against other modules. To test the integration of a module $\alpha$ against that of a module $\beta$, a stub of module $\beta$ is used in place of the actual module. This stub contains the interface of module $\beta$ without any of its functionality. Instead, the stub will have logic to verify the correctness of the arguments passed from module $\alpha$ and record the results. After the test iterations are completed, all the modules have analogous stubs; at this point, the system is considered fully tested for integration.

3 Assumptions and Exclusions

This section describes the assumptions and exclusions in the integration testing.

3.1 Assumptions

During the integration testing phase, we assume that each module’s internal logic and processes have been fully tested through unit testing. Integration testing only ensures the correct interfacing between modules.

3.2 Exclusions

Since the number of possible interaction sequences can be combinatorially large, we will evaluate the sequences and determine the most important ones for ensuring system correctness, security, and reliability.

4 Entry and Exit Criteria

The following subsections describe the prerequisites and conditions for each iteration of integration testing to occur.

4.1 Entry Criteria

Integration tests can only be initiated after all the following are completed:

- all modules for the particular iteration of integration test have successfully undergone unit testing
- stubs were implements for all participating modules
- consent of the Project Lead is obtained
- consent of the Test Lead is obtained
4.2 Exit Criteria

Integretion tests can be halted after any of the following are met:

- no defects or inconsistencies were observed after all test cases were run
- at least one defect or inconsistency was observed, which differs from the expected outcome of a test case

5 Participants

This section describes the roles and responsibilities of team members involved in the integration testing.

5.1 Roles and Responsibilities

For the integration test, the following roles are assumed by the following team members:

Test Lead Archit Baweja

Tester Archit Baweja, Drew Hall, Sunny Huynh, Kevin Lynch, Kanwarpreet Sethi

5.2 Reporting

Any problem discovered by a Tester must be documented and reported to the Test Lead. A problem report must be submitted to the Project Lead, and addressed during a meeting. After each iteration of integration testing, the Test Lead must compile a progress report and submit it to the Project Lead.
References
