

# Building Handheld Applications with VE

Prepared by  
Intelliun Corporation

## Contents:

- Overview
- Scope
- Development Artifacts
- Final Result
- Development Metrics
- Conclusion

## Overview

---

The Virtual Enterprise (VE) is a Java platform for the rapid development and deployment of n-tier applications. VE/Server layers on top of any third-party J2EE server, capitalizing on their robustness and scalability, yet radically simplifies and accelerates the development effort. VE/Designer provides visual modeling tools to capture the business logic in UML, and to personalize the look-and-feel for a variety of user access devices.

This paper provides an overview of a multi-client-access application built completely with VE, which was required to support web-browser access from a desktop (IE or Netscape) and a wireless handheld device (PocketPC). The primary objective of this paper is to:

- Describe a multi-client-access application,
- Illustrate the artifacts built with VE for developing this application, and
- Discuss the development metrics including effort and required skill sets.

## Scope

---

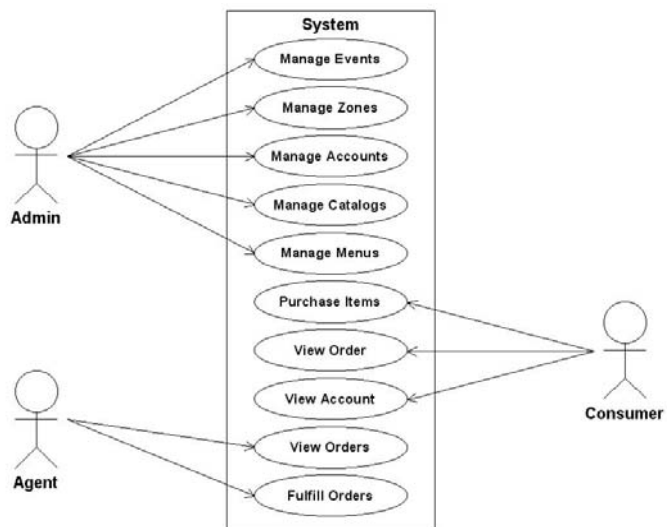
The target application is a fully functional application co-developed by one of Intellium's customers and Intellium's Professional Services for the purpose of bringing the Always-On Internet to live entertainment venues and sporting events. The application provides event attendees with wireless access to video instant replays, event statistics, and audio streams, as well as ordering concession items—all via a handheld device and without leaving their seats. This paper discusses the first iteration of the application, which focuses on product catalog management, venue and media event management, customer account management, and customer ordering and fulfillment.

The scope of this iteration contains 10 use cases and 3 actors as shown in the diagram to the right.

## Functional Overview

The application includes the following major functions:

- **Venue and Event Management:** the administrative function to enter and maintain customer venues (locations) and media events (such as a sporting event, musical performance or other type of event). For each venue, manage the menu of items that can be ordered from the product catalog, along with available sizes and prices.
- **Account Management:** the administrative function to capture and manage end-consumer account information, along with tracking their event attendance and order history.



- **Product Catalog Management:** the administrative function to define and manage product categories, product variations (e.g. size, color), and product information including images and icons.
- **Order Entry Function:** the consumers' order entry function to place food or product orders for in-seat delivery, along with managing the process of fulfilling consumer orders.

### Application Users (Actors)

The application is designed to support the needs of the following users (or actors):

- **Venue and Media Event Managers (Admin)** – the administrators of customer venue (location) and customer event information and processes, including the creation and maintenance of the product catalogs.
- **Service Managers (Agent)** – the managers for a specific event who are responsible for receiving consumer orders and fulfilling these orders by working with concession stands in their designated zone along with the delivery personnel.
- **Consumers** – the customers of the services provided through the handheld device at an event.

### Non-Functional Requirements:

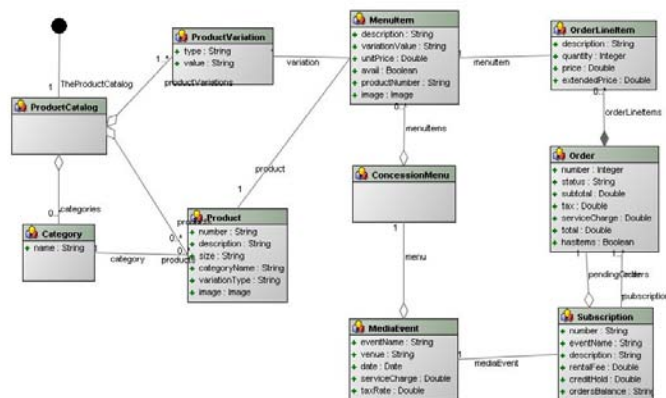
Each actor from the list above has a different access requirement. The Administrator must have off-site access to all of the system setup use cases, while the Agent requires on-site access to the event specific setup information and real-time orders. The Consumer is also on-site, but requires wireless access via a handheld device. Also, the user interface for the handheld device should not require the use of a stylus, which would impose additional constraints on button size, layout and the use of images, instead of text whenever possible.

In addition, there are the usual operational requirements including scalability, high availability and quick response time.

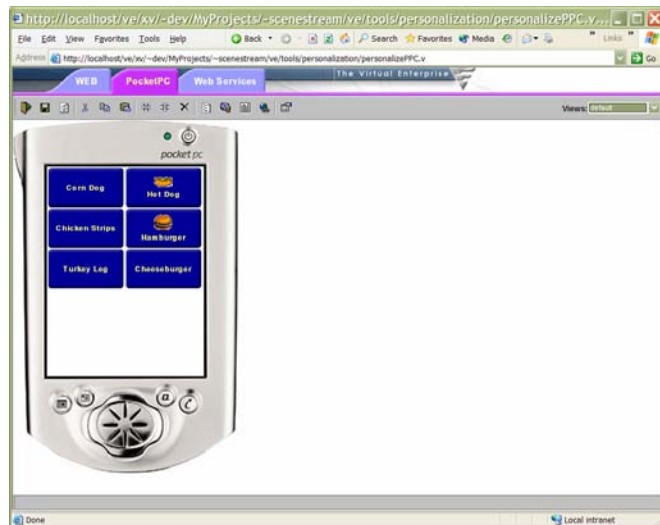
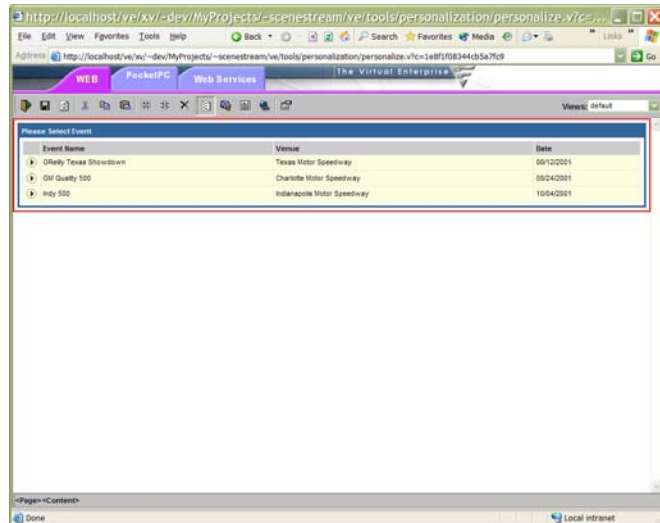
### Development Artifacts

The above uses cases were developed completely with VE using UML class and activity diagrams, along with the capture of the business logic to flush out the model. Three portals were created with VE, one per actor, and were personalized to meet their access requirements. The following is a more detailed list of the development artifacts:

- There are 18 business objects (Metaobjects or classes) with a total of 11 operations (very simple activity diagrams in this application). The object model to the right shows some of these objects along with their attributes and inter-relationships.
- There are 10 stand-alone processes (also very simple activity diagrams), one for each use case.



- There are a total of 21 activities distributed over the 21 processes (the 11 business object processes plus the 10 stand-alone processes), which yields an average of one activity per process in this application.
- The database schema and SQL statements were automatically generated by VE requiring no expertise with SQL, database administration or object-relational mapping.
- The user interfaces for both the desktop web browser and handheld browser were also automatically generated by VE, then tailored to the desired look and feel using the VE/Designer Personalization tool. No additional skill sets like HTML, CSS, JavaScript, JSP or Java were necessary.
- The resulting user interface consisted of 30 web pages, the majority of which were automatically generated by VE. The remainder were personalized using VE's WYSIWYG personalization tools.



## Final Result

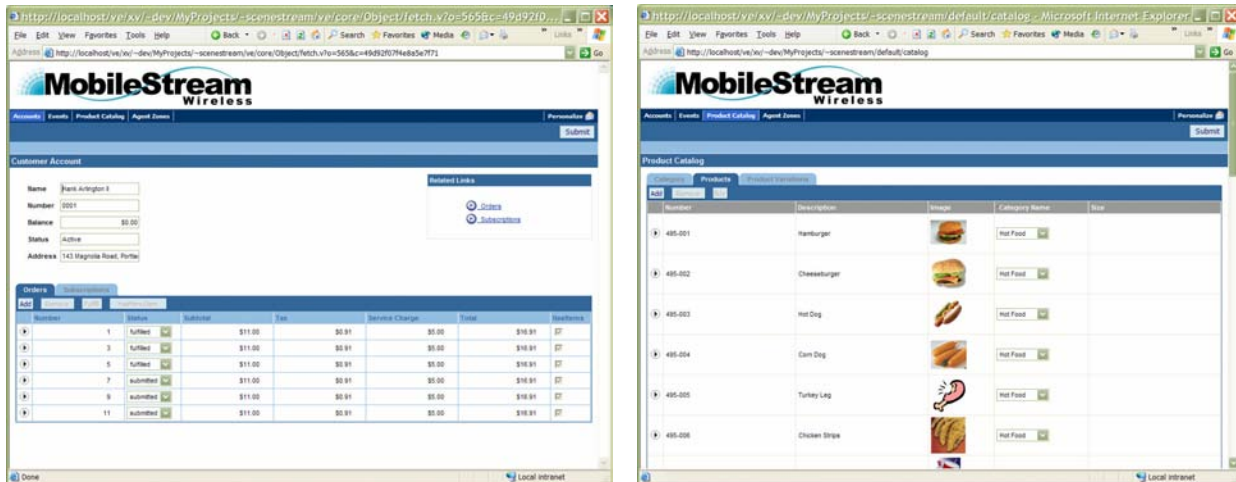
By the end of Iteration 1, the completed functionality constituted a fully functional n-tier application ready for deployment on any J2EE application server. VE-based applications can be deployed using VE/Server as pure Servlets (on Tomcat, as an example), or as Servlets/EJB (on leading J2EE application servers such as WebLogic or WebSphere). In addition, VE supports a wide range of relational databases (e.g. Oracle, DB2, MS-SQL, and Pervasive.SQL), and other persistence formats like flat files, XML documents, and LDAP servers. VE's architecture provides tremendous flexibility in deploying the same application logic on a variety of deployment configurations.

The following screen captures show the Purchase Items use case, accessed by the Consumer actor via a handheld device. In this sample from the use case, the Consumer peruses the menu categories before selecting one—in this instance *Hot Food*. The Consumer then selected *Hamburger* within the category before entering the desired quantity. Finally, the Consumer previews the order before pressing the *Checkout* or *Shop More* buttons.



Sample Handheld Browser Screen Captures

The following screens show an example of some of the back office or administrative use cases. These web pages required very little personalization. The sample screenshots below show web pages from the Customer Account Management and Product Catalog Management use cases.



Sample Desktop Browser Screen Captures

## Development Metrics

---

The total development time for this first iteration was 40 hours, and required the following skill sets:

- UML Class Diagram,
- UML Activity Diagram,
- Basic object-oriented design, and
- VE/Designer

The development effort using VE/Designer was preceded by an 8 hour scoping and analysis effort.

## Conclusions

---

This paper describes the requirements, approach and results for developing a fully functional application with dual client access devices, desktop and handheld. The 10 use cases described were developed in VE exclusively, and in just 48 man-hours from start to finish.

A similar exercise using traditional J2EE development tools would have taken several man-months of effort and would have required a significant number of resource skills, such as object-oriented design, UML, Java, Servlets, JSP, EJB, JDBC, SQL, HTML, JavaScript and CSS. The large number of required skills would have probably required a team of 2 to 3 developers. In addition, the developed artifacts would have consisted of hard-coded business logic in Java methods, database tables, and JSP and HTML pages, which are harder and more costly to maintain.

VE provides a platform for the development and deployment of n-tier applications providing automatic support for access via a multitude of thin-client devices. The application presented in this paper supports seamless concurrent accessibility from two platforms, handheld devices and desktop computers. Each platform has its own level of sophistication, screen space, usability and related issues that affect the tailoring of the application's user interfaces to the needs of the target user. In spite of this complexity, in 48 hours a fully functional application was developed to satisfy 10 use cases and 3 different actors.

The full development cycle included analysis, design, implementation and functionality testing. In addition, the development effort required only a few resource skill sets, and resulted in artifacts that are easy to understand and maintain. The demonstrated productivity improvements of the initial development also will apply as new functionality is added to future iterations or over the overall lifecycle of the application.