David Nimon

Principal Software Engineer

(469) 207-3876 david.nimon@gmail.com https://davidnimon.com
https://linkedin.com/in/david-nimon

SUMMARY

Software innovator and leader with over 10 years of full-stack, mobile, and embedded technology experience, with roles encompassing architecture, detailed design and code - usually, with a view towards developing or leveraging reusable frameworks.

EDUCATION

2014

Fort Worth, TX

B.S. Computer Information Systems
Tarleton State University

EXPERIENCE

2010 - NOW

Dallas, TX

MentorTek
https://mentortek.com
MentorTek is a contract development company providing software, hardware, signal processing, and data analysis resources with a vision to help shape the future of small to mid-size businesses. A sampling of my projects showcasing many architectures:

- Developed an extensive social media platform, including video-conferencing with document/desktop sharing.
- Managed a team of developers to build a remote patient vitals monitoring solution. Developed 3rd-party health system API integration, supportive of doctor to patient communication.
- Developed migration strategy for a customized Shopify store from their existing 1.0 platform/themes to a new 2.0 theme. Assumed responsibility for development and maintenance of their current production store.
- Served as manager, architect, and developer for a cloud focused Enterprise Resource Platform (ERP). Provided an industry unique No-Code/Low-Code method of ERP user workflow creation.
- Served as manager, architect, and developer for a cloud based Content Management System designed to control remote marketing media displays along with the embedded application required for the media display units.
- Managed a team to build an MVP for the first cloud solution for online life insurance sales that provided real time underwriting.
- Developed a framework and set of mobile applications targeted for large experiential events. Supported onsite operations for Superbowl New York City.
- Developed a client application for a popular commercial VPN product supporting iOS, Android, Windows and Mac - all sharing common code for UI and VPN services.
- Developed e-commerce payment gateway integrations as part of various projects, including Stripe, PayPal, CardConnect, Authorize.Net.
- Managed a team and developed a cloud service that provided a drag/drop method of creating mobile applications. Developed automations for publishing the applications created by the service into the Apple and Google stores, using Fastlane for automation.

RESPONSIBILITIES

Architect, design, code, and debug new software applications

Maintain, upgrade and improve the performance of existing software

Recruit, train and supervise software engineers

Manage project client relationships and their project requirements

Contribute to and/or leverage relevant open source projects

SKILLS

Frameworks

Angular & Material | Node & Express | Docker | Ionic | Laravel | Flask | React | Xamarin | .NET & .NET Core | Wordpress | Custom

Services & Integrations

AWS | Shopify | Heroku | MongoDB Atlas | Rackspace | Sendgrid | Twilio | Firebase | Payment Gateways | Redox & Xealth | Azure Notification Hubs | Klaviyo

Languages

Javascript/TypeScript | PHP | Python | C# | Java | Rust | C++ & C | Objective-C | Swift | Ruby | HTML & CSS

Databases

Postgres | MySQL | MongoDB | SQLite | ORM and Query Builders

Project Tools

JIRA & Redmine | Lucidchart | Slack | Mattermost | Webpack | CMake

Development Tools

Git | Visual Studio Code | Visual Studio | vi and nano | Eclipse | GitHub Actions | Puppeteer

IT Administration

Managing AWS Instances | Managing physical systems | Database management | VPN remote management | Small business network setup | OS GUI & Terminal

Leadership

Collaboration | Communication | Planning | Architecture | Training | Agile Development | Object Design

This site represents a portfolio of work attributed to me, David Nimon.

I am a software innovator and leader with over 10 years of full-stack, mobile, and embedded technology experience, with roles encompassing architecture, detailed design and code - usually, with a view towards developing or leveraging reusable frameworks.

I am a Principal Software Engineer at a contract development company called MentorTek founded in 1996 to provide software, hardware, signal processing, data analysis, and patent application resources with a vision to help shape the future of small to mid-size businesses. I joined them in 2010, prior to completing my college degree at Tarleton State in 2014 (B.S. Computer Information Systems).

My primary responsibilities in that role include:

- Architect, design, code, and debug new software applications
- Maintain, upgrade and improve the performance of existing software
- Recruit, train and supervise software engineers
- Manage project client relationships and their project requirements
- Contribute to and/or leverage relevant open source projects

This portfolio contains my commercial projects, technology related skills, patent experience, and non-commercial educational/philanthropic endeavors organized into sections as described below. Links are also provided to a concise resume and as well as a complete document compilation of the entire portfolio in a single downloadable PDF file.

Sections

A project section with details depicting my specific roles along with functional overviews

A technology section illustrating the frameworks, commercial cloud services, 3rd party integrations, programming languages, databases and project management tools I used for these projects

A patent section with a discussion of MentorTek's patent authoring services along with my relevant contributions

A resume section with my resume and contact information

This section summarizes a set of my MentorTek commercial projects and a set of special non-commercial projects. Because of non-disclosure agreements with the project clients, the descriptions here are fairly high level and do not disclose UI/UX, backend, and other detailed design patterns.

Commercial Projects

Shopify Store Custom Development
Personalized online store for women's pants

Online store specializing in women's pants. A major feature of the store is a quiz that the customer takes that determines their size and shape for ordering product. I help maintain an existing Shopify 1.0 store and have been working on a new Shopify 2.0 version of the store.

- Assumed responsibility of improvement/maintenance of client's customized production Shopify 1.0 theme
- Developed a migration path for the client's store to Shopify 2.0 framework and selection of appropriate themes (selected 2.0 themes over Shopify's support for a headless implementation)
- Built and maintain the PHP backend for a fit finding quiz. The quiz results are stored in an MySQL database, currently with over two million records
- Built an extended version of the same quiz that also integrates with the API of a 3rd party photo capture/analysis service to obtain accurate measurements and retrieve/store the photos in an encrypted AWS S3 bucket
- Built a Shopify 2.0 theme extension with a new quiz user interface using bootstrap and HTML to present a step-by-step wizard for the questions with validation
- Built an integration of the new quiz with Amazon Mechanical Turk, to help automate data collection/analysis used to improve the fit algorithm, supporting both cross-sectional and longitudinal studies
- Improved the integration of the marketing service Klaviyo, to help streamline the process of sending emails when a quiz was taken and to track more data about the customer for future emails
- Designed solutions for issues with the 3rd party CMS portion of the client's store

Conferencing

Social platform with video conferencing

An extensive social media platform, with friends, messaging system, groups and a personal profile that is managed with a custom drag/drop editor. It has an easy to use video conferencing feature set, using your browser or provided desktop and mobile apps.

- Built original version for client several years ago using Flash framework, along with a PHP and Adobe Flash Media Server backend, while also supporting the open source Red5 video server
- Created a new version built with Angular frontend, Node backend, Postgres database, Ionic Framework for mobile and desktop apps, and OpenVidu server leveraging WebRTC for the video conferencing
- Created a theming engine for the frontend, separating the logic of the page from it's presentation. This was used so that browser and mobile could share logic but have different user interfaces, bootstrap and angular material for desktop, Ionic Framework for mobile and desktop
- Created for video conferencing an industry unique capability to share documents and/or on-demand video streams, resulting in higher quality with lower bandwidth demand compared to typical desktop sharing features (desktop sharing was also included)
- Ionic Framework was used to build mobile and desktop apps, using that framework's user interface modules and build process - supported sharing interface logic between desktop, mobile, and browser environments leading to much faster development cycles compared to full native applications
- Created mobile browser integration to support launching the app directly into a conference
- Created an Express JSON API subsystem, with support for running longer running functions, such as document conversion and email/SMS messaging using an AWS Lambda job technique
- Integrated ffmpeg for converting the videos into a common format and extracting screenshots, storing the results in an AWS S3 bucket
- Created a Docker Compose environment for all of the services required to run the project, used in Amazon EC2 instances for development and production

AWS to Self-Hosting

Migrate services from AWS to onsite servers and database

Our company was charged for multiple AWS EC2 Instances, so I wanted to work on saving money. To that end, I created a local server cluster. I installed Pfsense on a PC to act as the router and setup Docker containers on an Intel NUC and a Raspberry Pi. This helped me learn more about Docker Compose and publishing services hosted locally with Cloudflare.

- Installed Pfsense on a PC and setup a Ubiquiti AP for WiFi
- Installed Ubuntu Server on an Intel NUC and a Raspberry Pi, then installed Docker and Portainer for container management
- Created Docker Compose environments for Gitea, Mattermost, Redmine, and KeyCloak, and an authorization server - migrating from the original on AWS as EC2 instances, without docker
- Setup all of the above services to authenticate with KeyCloak, so that only one login is required
- Setup Cloudflare tunnels to have the services accessible on the internet, without exposing the IP address
- Setup a backup process automatically uploading all onsite data to an S3 bucket

Healthcare
Remote patient monitoring

Health company in the remote patient monitoring space, with integrations with various bluetooth devices to track vitals.

- Led a team of in-house developers within the company along with an outsourced overseas team, functioning as a liaison with the stakeholders (customers, product managers, and the management team)
- Managed issues and reports through JIRA, and used Github to help manage the code development
- Built the service backend using Node & Express, with a database in MongoDB Atlas
- Built the service frontend using Angular with Bootstrap and Angular Material components
- Designed and built a job queuing system with associated API using Node forks and RabbitMQ to store the jobs
- Built an API query validation framework
- Built integrations with Twilio for sending text messages, automated calling when an alert had occurred, and video conferencing
- Developed a system for inviting a 3rd party doctor to a two way video call with a patient, initiated via a push notification
- Developed a set of APIs to support 3rd party queries retrieve customer health vitals measurements
- Created a system for Github CI Actions (pull request creation/changes) automated tests of the backend, using Mocha and a in-memory MongoDB server
- Built an application to run on a Linux-based bluetooth hub, using C++, integrating an MQTT client and a bluetooth device layer
- Created a portal user interface to allow customer support to configure and manage the bluetooth device hubs
- Managed the Heroku servers used for both production and staging
- Developed integration for multiple health providers, using solutions including Xealth and Redox to integrate with Epic and other systems, providing patient information and doctor portal single sign-on services

Enterprise Resource Platform
Browser-based ERP application with drag/drop Low-Code/No-Code UI/UX

The ERP application was built to be completely web-based and be user friendly, as an alternative to the older systems available on the market. It interfaced to multiple devices, including printers and scanners.

- Selected Angular for the frontend, Node & Express for the backend, and Postgres for the database
- Built generic widgets using Angular components such that all UI elements and server API calls were expressed in a JSON format, so that backend developers could build workflows without any Angular knowledge
- Created a drag/drop editor for the JSON files, allowing someone with no technical knowledge to build custom table view pages and workflows
- Developed a docker environment for the frontend and backend to run on, supporting direct USB device access through the docker interface
- Evolved docker environment for Intel NUCs, including support for USB device hotplugging
- Created a TypeScript-based class structure for each device for device API command abstraction
- Developed interfaces for label printers and document scanners, installing the drivers into the Linux docker environment, and then creating an interface to list, manage, and print to the requested printer. I researched existing Linux commands for managing printers, deciding on lp. I then added this to the node backend so that when a user wanted to manage printers or print a document, an API call would occur that would use this interface
- Used lp for printer interfaces (list, manage, and print) and scanimage for document storage/retrieval functions
- Built a report generation tool, based on the same JSON-based format using the Puppeteer tool to create a PDF - reports were generated reusing same UI and logic that generates the web pages
- Developed interfaces with network-based cash drawers, implementing a configuration system to add a new cash drawer, and then implemented functions to open and check the status of the cash drawer
- Created a Postgres migration builder, using models built in Node to automate the creation of migrations

E-commerce Integration
Interfacing with CardConnect credit card terminals

The client had a Point of Sales system without an integration to CardConnect terminals. I built a socket-based C# application to communicate with the Point of Sale System and the CardConnect API.

- The application was built with C# using .NET Core
- The POS system would transmit a request over the socket to take a credit card which would trigger the CardConnect interface
- The CardConnect device interfaced with REST API calls, using multiple servers - each server supporting different credit functions
- Created a socket protocol whose payload contained the combination of a message header and a JSON representation of the CardConnect transaction
- Developed an object class structure for communicating using REST APIs abstracting the detailed differences between multiple servers
- Provided a JSON-based, multi-terminal configuration system, to allow the user of the application to run multiple terminals
- Used terminal API to capture customer signature images
- Built a PowerShell script that compiles, builds, and packages the application into a single ZIP file
- Provided detailed documentation for setup and use of the application, along with a GUI-based sample application
- Provided client support, with additional improvements and issue resolution

Experiential Marketing Applications
CMS to control marketing displays remotely

Helps business with storefronts market their products through the use of various types of electronic displays, such as projectors or mounted televisions. I helped build a CMS and the application running on the control unit for the displays.

- Managed a team of developers working for MentorTek to build a media CMS portal, functioned as liaison between various stakeholders and the development team
- Developed backend architecture using Laravel, a framework for PHP in conjunction with a frontend using React & React Router, coupled with Bootstrap components
- Built a React component wrapper around the React Data Table component, improving upon tech debt by reducing code duplication
- Built a conversion system for videos uploaded by customers to run on the displays controlled by the CMS using two architectures - A python script designed to run FFmpeg & pushing the result to a S3 bucket, switching to AWS Elemental MediaConvert for increased performance
- Designed the application running on the Linux-based control unit managing the media displays, initially using Electron, with HTML/Javascript and Node components
- Developed a second version of the control unit application, using a node backend and an off the shelf Chrome instance, all running on the control unit, supporting GPU rendering and using a fault tolerant WebSocket communication protocol between the two layers
- Designed a novel communication method between the CMS server and the media control unit that replaced a RabbitMQ server with a reliable protocol based on using pair of S3 buckets which provided improved scaling and performance
- Built an automation subsystem to manage (create and deploy) OpenVPN profiles to support remote debugging of thousands of media control units via SSH and remote login
- Migrated CMS deployment from direct component installations (e.g., PHP, gulp) on the EC2 to a newer version that used Docker Compose on the EC2 instance

NoCode Platform
Build mobile apps via drag/drop

Service that lets small businesses create a mobile app and have it published in the store without having to write any code, providing a friendly user interface to create content and configure their app.

- Managed a team of developers working for MentorTek to build a portal and mobile app framework
- Used Meteor for the initial backend and frontend architecture, along with Xamarin for the mobile app framework
- Built the mobile applications with a combination of Xamarin C# with a Webview running a mobile-friendly local copy of a portion of the portal code
- Developed a push notification system as a tie between the mobile application and the backend using Azure Notification Hub coupled with a custom notification scheduling system
- Built automations for publication of the mobile apps to the Apple Store (iOS) and the Google Play Store (Android)
- Designed a ruby script based on Fastlane running on a Mac Mini to build and publish to the Apple Store
- Developed an interface on a Windows AWS EC2 Instance, to accept build requests and publish to the Google Play Store
- Managed a team of developers to migrate the feature set from Meteor to improve performance and development processes using a custom UI and backend framework based on Node Express

Online Life Insurance Portal
Quotes for life insurance made in real time

Company requested a interactive branching form to help the customer find their best life insurance.

- Managed a team of MentorTek developers to build the frontend and backend of an MVP web application to query potential customers providing them with real time life insurance quotes
- Developed an architecture using Python Flask, SQLAlchemy, Postgres, and React
- Developed custom interfaces for 3rd party insurance underwriting and accounting APIs (SOAP/XML, etc.)
- Supported client's effort to build an in-house, post-MVP team resulting in a successful transition

Experiential Marketing Applications
Onsite real time event management and data collection

Many interactive marketing tools and events with multiple technologies involved.

- Developed a set of mobile applications, using Xamarin for Android and iOS, and a CEF-based Windows application
- The UI for the applications was a Webview whose Javascript code was supported by generic native mobile backend C# code - UI Javascript had the majority of the control logic
- Implemented an interface to a USB device to read RFID tags for the Windows application
- Developed a Linux-based application to control microprocessor custom towers, creating methods for remote deployment
- Designed custom tower application support for both UHF (used for ranging) and HF (used for tap control) tags
- Developed custom tower features for onsite touch screen gamification/interaction
- Built a system for tracking dwell time of people in an event via BLE tags as they came in and out of range
- Developed a green screen control app, managing an off the shelf video composite application
- Built an FFmpeg based python script to run on the backend servers to take a set of videos and efficiently combine them
- Built a portal to allow for the marketing team and customers of the company to manage their content without needing to change code or build a new app - the backend for the portal was Python Flask with SqlAlchemy & Postgres, with a React frontend
- Managed a team of MentorTek developers to add new features and update the portal, serving as a liaison among all stakeholders
- Helped manage on-site operations for New York City Superbowl experiential event
- Maintained a legacy iOS Objective-C app which included device driver interfaces such a plug-in driver's license scanner

VPN Remote Management/Security
VPN client to connect with self-hosted servers with easy to use service option

Company provides a fully featured open source VPN client and server, along with offering a paid service for access to provided VPN servers.

- Developed the user interface of a VPN client application, supporting mobile devices, Windows, and Mac
- Built to the backend server API specifications for adding account status, bookmarks, and available servers
- Supported the backend development team for new features and diagnosing issues
- Designed a common C++ layer for all of the platforms to drive the UI interface, eliminating most platform-specific code
- Implemented the CEF interface for the UI for Windows and Mac, while maintaining the platform specific Objective-C code needed for iOS and Mac, the Java code for Android, and the C++ code for Windows
- Implemented a JNI layer to allow the android application to use the common C++ interface logic
- Handled creating and publishing releases of the application with test builds to TestFlight for iOS and publication on a shared drive for the rest of the platforms

R Shiny Server
Setting up and maintaining R Shiny Server

Shiny allows for interactive web apps to be built through R. Client wanted a Shiny server accessible from the web.

- Given an existing Shiny application, determined the necessary dependencies
- Created a Docker environment that installed R and shiny, then installed the necessary R package dependencies
- Helped support maintenance of Shiny application and updated to handle Linux environment, compared to the original Windows environment it was developed with
- Created an example Shiny app that used the Web File System Access API to retrieve directory listings and files in those directories.

Conference Registration Website
Maintained website and built registration system

I built and maintained a Wordpress-based website for the organization. I also built and maintained a custom PHP based registration system.

- Installed and updated Wordpress plugins
- Updated the theme and pages due to requests
- Created registration system for customers to fill out form, pick add-ons, and pay registration fees, through a PayPal interface
- Built an administration portal that allows the organization to add or modify products, change the content of the registration page, and see customer transactions
- Built a label generation page for bulk or individual customers for creating badges for the conference, using PHP and HTML

Non-Commercial Projects

Luxtorpeda
Linux app for using source ports with Steam

Older games that have been released on steam can include out-of-date binaries which can be difficult to run on newer systems. This project allows for Linux users to download better versions of the application part of the game, using source ports that expand on the original game source code, without having to go through manual steps. I am the maintainer of this open source project, which was forked from an inactive project.

- The project includes a user interface and a repository of build scripts for the various ports
- The user interface is built in Rust, which shows the user the available ports, then automatically downloads the latest version, configures, and launches it
- Created bash scripts for building the ports. These scripts handle downloading the source code from git repositories, building & configuring any dependencies, then configuring build tools, such as CMake or make, to build the port
- Created GitHub CI Actions to automatically build the UI and port projects when a pull request or commit is pushed. The CI process will then create GitHub releases for storing the build artifact
- Handled issues and pull requests from users of the project
- The latest version of the application has been downloaded over 2000 times
- Contributed a pull request to the egui project for a new feature
- Contributed additional pull requests to a game engine project, to improve the handling of the window and mouse on Linux

Tower Defense

Tower defense game built with Godot

I built this game to learn about the Godot game engine. I used open source assets and followed tutorials on how to use Godot, then expanded on the features.

- The game logic is written in GDScript, which looks similar in syntax to Python. I learned about signals, modifying and creating UI elements, and modifying and creating game objects
- Added features to the game so that the settings of the objects in the game, like the turrets and enemies, can be customized in-game with an editor
- Added music that changes based on if idle or in a wave, and then added sound effects for button clicks, explosions, etc
- Added an additional laser turret, and a missile turret with fired missiles, which helped me to learn more related physics object

Non-Profit Website
Website maintained for non-profit

I took over maintenance of a Wordpress-based website for a non-profit.

- Installed and updated Wordpress plugins, including SEO and caching, to improve searching and performance
- Updated the theme and pages due to requests
- Built Xamarin Android & iOS Apps for a video catalog, to allow videos about the non-profit to be downloaded to be shown later without internet access. Built another version as a PWA in Javascript and HTML

This section provides a technology summary pertinent to the development of the projects listed in this portfolio, with an emphasis on only those that had major project impact and led to a deep understanding.

Custom Frameworks

Custom frameworks built using a combination of open source projects and custom code, used as a basis for projects

Lightweight website stack based on PHP and MySQL

Frameworks designed to simplify and localize the representation of domain knowledge using Node and Frontend Javascript, supporting PostgreSQL and MySQL. One framework was based on shared common objects between the frontend and backend, while the other confined all domain knowledge in the frontend, which greatly minimized the API footprint

Framework using Python and OpenCV to support image capture and Optical Character Recognition services

Custom embedded device framework built with Chromium and Node, abstracting CMS server and serial device communication used to manage kiosk media displays

Framework for Node Express using a Typescript library abstracting USB embedded device communication/control

Reusable, lightweight abstractions of FFmpeg interface allowing Linux servers (Node and Python) to support video stitching, conversion, and chroma key extraction - including a custom C++ FFmpeg filter based on an existing chroma key library

Linux-based BLE hub framework, written in C++ which integrated libraries for MQTT clients and Bluez Bluetooth communication

Angular based abstraction for code sharing of frontend event processing logic between responsive Bootstrap and Ionic mobile themes

Commercial and Open Source Frameworks

Commercial and open source frameworks used in projects to build custom cloud and mobile applications

Angular with Router, Material, and Bootstrap

Node with Express & Mocha for Creating Tests, with Knex, Sequelize, or Mongoose

Docker with Docker Compose

Ionic Framework with Capacitor

Laravel

Python Flask with SQLAlchemy

React with React Router and Bootstrap

Xamarin Android & Xamarin iOS

.NET & .NET Core

Meteor

Wordpress

Services

Commercial services used to deploy major features or entire infrastructure for applications

AWS - EC2, S3, Lamdba, CloudWatch, VPC, RDS, Mechanical Turk, Elemental MediaConvert

Shopify

Heroku

MongoDB Atlas

Liquid Web

Rackspace

Integrations

3rd party services integrated to extend applications

Sendgrid

Twilio - Text Messaging, Calls, Video Conferencing

Firebase Cloud Messaging

Payment Gateways - Stripe, Paypal, Authorize.Net, CardConnect

Redox and Xealth for HL7 health system integration

Insurance ceding/fronting and reinsurance APIs for underwriting and accounting

Support for 'Know Your Customer' and 'Customer Identification Program' - LexisNexis (EIR, Credit MVR, Risk Classifier), MIB Consumer File, Milliman RX, Ingenix MedPoint

Azure Notification Hubs

Klaviyo

Languages

Expertise in multiple high-level programming languages

Javascript (Node & Browser-based) & TypeScript

HTML & CSS

PHP

Python

C#

Java

Rust

C++

C

Objective-C

Swift

Ruby

Databases

Creating and managing schema along with SQL, NoSQL, and ORM code development

Postgres

MySQL

MongoDB

SQLite

ORM and Query Builders - Sequelize, SQLAlchemy, Knex

Project Tools

Tools used to help manage projects, including team collaboration on code and design

JIRA and Redmine, managing issues and project progress

Lucidchart, supplementing technical documentation including UML and architecture/database diagrams

Team collaboration, including Slack and Mattermost

Build tools, including CMake and Webpack

Development Tools

Tools used to help development

Git command line and GUI tools, such as Sublime Merge or Sourcetree. Created, managed, and reviewed Github pull requests

Visual Studio Code

Visual Studio

vi and nano

Eclipse

CI such as GitHub Actions

Puppeteer

IT Administration

Knowledge in server and infrastructure management

Managing AWS Instances with Linux, installing packages or Docker environments, and maintenance

Managing on-premises physical systems with Linux-based servers

Using AWS Security Rules and utilities like iptables to secure servers

Setting up and managing databases for production (Linux servers) and development platforms (Windows, OS X, Linux)

Management of embedded systems using Intel NUCs, Raspberry PIs, and custom BLE devices

Development of remote management methods for systems using VPN servers, providing profile deployment automation

Setting up and maintaining small business on-premises networks, using pfSense for the router

Using Windows, OS X, and Linux in a GUI environment. Using Windows Command Line & PowerShell, OS X Terminal, and Linux Terminal

This section summarizes patent applications to which I co-authored and/or contributed relevant technology. MentorTek has authored over a dozen on behalf of various clients, of which I contributed to the ones listed here.

METHOD AND SYSTEM FOR IMPROVED APPAREL FIT

A system and method implemented by a computer to determine the correct garment fit, where a fit is represented as a label which includes shape and size, and is suitable for mapping to garments manufactured appropriate to that representation. The system provides a user interface including a perception interface (presents and enables selection of user fit perception choices) and measurement interface (presents and enables options for capturing measurement data), measurement analysis server, fit analysis server. The measurement analysis server receives from the user interface running concurrently data communicated and analyzes the data to produce a measurement representation. The fit analysis server receives from the user interface running concurrently data communicated and analyzes the data to produce a fit representation. The method provided includes the steps of: (i) acquire perception representation, (ii) acquire measurement representation related to specific body Points of Measurement, and (iii) calculate fit representation using perception and measurement representations

PLATFORM AND RELATED PROCESS FOR PROVIDING PUSH NOTIFICATIONS VIA UNIQUE SUBSCRIBER APPLICATIONS

The disclosed principles provide a platform providing subscribers the ability to send custom push notifications. The platform is application-based, using mobile applications for use on a mobile device. Subscribers are provided a subscriber-level application from the platform. That application could be customized for the subscriber, or the subscriber may have a given amount of customization in order to personalize their own application. Users of a subscriber's unique application download the application via the platform. The subscriber will have administrative access within that application, via user name and password, and can coordinate to whom push notifications may be sent. The subscriber can designate individual targets to receive those push notifications, or may create groups for receiving the push notifications. Users of that subscriber's application could be allowed to shop for the subscriber's products via their application, and can receive push notifications sent via the platform.

POINT OF SALE ADJUNCT CONTROL

An adjunct control system includes a point of sale workflow interface through which visual aspects of a point of sale workflow are received, an adjunct controller that processes the received visual aspects of the workflow to detect a transaction type and a transaction amount, and a credit card terminal interface through which the controller interacts with a credit card terminal to complete a transaction captured in the workflow.

MULTIPLE CAMERA GROUP COLLABORATION SYSTEM AND METHOD

An improved group collaboration system providing control and management of multiple cameras for multiple users (members). The system includes a server for system management and a server for video data. Multiple clients, each having multiple cameras, may access the system simultaneously. Each client has a browser interface that allows a user to access multiple member cameras. Each selected camera is displayed simultaneously in the selecting user's interface. A bandwidth distribution algorithm provides quality of service for the multiple camera feeds.