Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Marcin Przekop
Nr albumu: 214757

Aplikacje na urządzenia mobilne typu smartphone

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
dr Janiny Mincer-Daszkiewicz
Instytut Informatyki

Wrzesień 2007

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego praca

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.
Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.
Oświadczam ponadto, ze niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora pracy

Streszczenie

W pracy przedstawiono projekt i implementację aplikacji do zarządzania listami zakupów napisaną do wykorzystania na urządzeniach mobilnych typu smartphone. Rozwiązanie obejmuje komunikację klient-serwer pomiędzy telefonem, a serwisem internetowym za pomocą technologii usług sieciowych. Bezpośrednia komunikacja pomiędzy telefonami odbywa się na zasadach P2P z wykorzystaniem krótkich wiadomości tekstowych (SMS). W pracy omówiono również budowę standardowego smartphona oraz dostępne mobilne systemy operacyjne.

Słowa kluczowe

urządzenie mobilne, smartphone, SMS, .NET Compact Framework, Wi-fi, Windows Mobile 5.0, technologie mobilne, aplikacje mobilne, usługi sieciowe

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka

Klasyfikacja tematyczna

H. Information Systems
H.4 Information Systems Applications
H.4.3 Communications Applications

Tytuł pracy w języku angielskim

Mobile Applications for smartphone

Spis treści

Wprowadzenie

Upowszechnienie się komputerów osobisty klasy PC otworzyło przed ludźmi możliwości dostępne wcześniej tylko dla dużych korporacji i instytucji rządowych. Pojawiły się nowe typy aplikacje do zastosowań domowych. Wszystkie te programy były jednak przeznaczone na stacjonarny komputer i wykorzystywały zasoby tylko jednej maszyny. Pojawienie się Internetu wymagało całkowitej zmiany podejścia do tworzenia aplikacji. Komputer przestał być odizolowaną jednostką i stał się elementem sieci. Od razu pojawiły się aplikacje wykorzystujące nowe możliwości przesyłania i współdzielenia informacji za pośrednictwem sieci komputerowych. Obecnie obserwujemy pokonanie drugiego ograniczenia nałożonego na komputery stacjonarne. Urządzenia komputerowe nie są już na stałe przypisane do konkretnej lokalizacji. Dzięki ogromnemu postępowi w sferze bezprzewodowej transmisji danych, miniaturyzacji procesorów i nośników pamięci oraz rozwojowi sieci komputerowych ludzie już niedługo będą mieć dostęp do potrzebnych im informacji „w dowolnym momencie, w dowolnym miejscu i z dowolnego urządzenia”[2]. Telefony komórkowe, a szczególnie smartphony, odegrają dużą rolę w realizacji tego założenia.

Urządzenia mobilne to nie tylko miniaturowe wersje stacjonarnych komputerów i ich wykorzystanie nie powinno ograniczać się do sprawdzenia grafiku, poczty elektronicznej, czy wiadomości podczas przemieszczania się. Możliwość nieograniczonego dostępu do informacji w dowolnym miejscu umożliwia budowanie systemów informacyjnych spersonalizowanych nie tylko dla konkretnego użytkownika, ale również dla konkretnego miejsca. Wymaga to od projektantów aplikacji zasadniczej zmiany dotychczasowych przyzwyczajeń.

W niniejszej pracy zaprezentowano aplikację przeznaczoną do uruchomiania na telefonie komórkowym nowej generacji – smartphonie. Głównym zadaniem aplikacji jest zarządzanie i przechowywanie list z zakupami. Program realizujący tę funkcjonalność nie miałaby racji bytu na komputerze stacjonarnym, z którego ciężko byłoby korzystać podczas pobytu w sklepie. Znakomicie sprawdza się natomiast na telefonie komórkowym, który z założenia zawsze mamy przy sobie.

Rozdział pierwszy zawiera omówienie podstawowych pojęć związanych z technologiami mobilnymi. Ukazano różnicę pomiędzy tradycyjnym telefonem komórkowym a smartphonem. Zaprezentowano budowę standardowego smartphona i dokonano przeglądu dostępnych na rynku mobilnych systemów operacyjnych i aplikacji. W rozdziale drugim znajduje się wizja przygotowanej aplikacji oraz przyjęte założenia funkcjonalne i niefunkcjonalne, jakie powinna ona spełniać. W rozdziale trzecim i czwarty znajduje się projekt i sposób implementacji poszczególnych elementów systemu, oraz dokładniej omówiono najciekawsze przypadki użycia. Rozdział piąty zawiera wyniki testów wydajnościowych aplikacji prezentując jednocześnie możliwości urządzeń mobilnych. Praca kończy się podsumowaniem, w którym przedstawiono dalsze możliwości rozszerzenia aplikacji.

1. Urządzenia mobilne

1.1 Pojęcia

W literaturze technicznej istnieje wiele różnych definicji terminów związanych z technologiami mobilnymi. Często definicje te bardzo się od siebie różnią. W tym rozdziale wyjaśnione zostaną podstawowe pojęcia związane z technologiami mobilnymi. Przedstawione zostaną różnice pomiędzy tradycyjnym telefonem komórkowym a smartphonem.

1.1.1 Technologie mobilne

Termin *technologie mobilne* odnosi się do szerokiego zakresu urządzeń i operacji związanych z dostępem, przechowywaniem i przetwarzaniem informacji z wykorzystaniem przenośnych urządzeń komputerowych takich jak laptop, PDA (Personal Digital Assistant), telefon komórkowy, tablet itp.

Wyróżnia się dwa tryby pracy urządzeń mobilnych[14]:

- Tryb podłączony (online) – urządzenie udostępnia przynajmniej jeden typ komunikacji bezprzewodowej (Wi-fi, Bluetooth, łącza podczerwieni (IrDa), GPRS). W trybie podłączonym aplikacje mobilne mają bezpośredni dostęp do zewnętrznych i zdalnych źródeł danych, innych urządzeń mobilnych lub stacjonarnych systemów komputerowych.

- Tryb odłączony (offline) – urządzenie ma bezpośredni dostęp tylko do lokalnie przechowywanych informacji (np. książki adresowej). Dane te mogą być jednak synchronizowane z innymi urządzeniami w czasie krótkich sesji komunikacyjnych. Wymiana danych podczas synchronizacji może następować w obu kierunkach.

1.1.2 Komunikacja bezprzewodowa

Tradycyjnym sposobem na połączenie dwóch urządzeń komputerowych jest wykorzystanie specjalnych kabli do transmisji danych. W przypadku komunikacji bezprzewodowej zamiast kabli wykorzystywane są fale radiowe. Jednak sam fakt wykorzystywania przez urządzenie komputerowe bezprzewodowych metod komunikacji nie oznacza, że dane urządzenie może być zakwalifikowane jako mobilne. Niektóre technologie bezprzewodowe, zaprojektowane jako alternatywa dla kabli, mogą być efektywnie wykorzystywane tylko wtedy, gdy jeden lub nawet wszyscy uczestnicy komunikacji w momencie transmisji sygnału pozostają nieruchomi lub poruszają się w bardzo ograniczonym stopniu. Z taką sytuacją mamy do czynienia na przykład w przypadku technologii opartej na poczerwieni lub wykorzystującej sieci bezprzewodowe WiMax.

W definicji komunikacji bezprzewodowej główny nacisk kładzie się na wykorzystanie fal radiowych jako nośnika informacji i metody tworzenia sieci komputerowych, a technologie mobilne wykorzystują niektóre z bezprzewodowych metod komunikacji w tworzeniu aplikacji i usług.

Szybki rozwój technologii mobilnych i bezprzewodowych metod komunikacji przyczynił się do powstania wielu nowych aplikacji i systemów komputerowych wykorzystujących komunikację głosową, przesyłanie wiadomości i transfer danych. Rosnąca liczba ogólnodostępnych sieci

bezprzewodowych i nowe usługi operatorów sieci komórkowych sprzyjają rozwojowi urządzeń mobilnych, będących niemal zawsze w zasięgu sieci umożliwiającej pracę w trybie podłączonym lub synchronizację danych z systemem zewnętrznym. Nowoczesne metody komunikacji oferują wysoki transfer danych, a przełączanie się urządzenia mobilnego pomiędzy punktami dostępu i różnymi metodami komunikacji odbywa się automatycznie bez udziału użytkownika.

1.1.3 Smartphone i tradycyjne telefony komórkowe

Początkowo termin *smart phone* wykorzystywany był w celach marketingowych w odniesieniu do nowych modeli telefonów komórkowych, które umożliwiały przechowywanie i przetwarzanie informacji w znacznie większej skali niż wcześniejsze modele. Jako dodatek do standardowej funkcjonalności umożliwiającej komunikację głosową i tekstową, smartphony udostępniały proste aplikacje do przechowywania danych osobistych oraz zarządzania czasem (kalendarze, notatniki, książki adresowe itp.). Pierwsze modele umożliwiały tylko bardzo ograniczoną formę komunikacji bezprzewodowej, najczęściej polegającą na bezprzewodowym podłączeniu urządzenia do stacjonarnego komputera. Obecnie wszystkie te funkcje (a nawet więcej) zapewniają standardowe telefony komórkowe.

Tabela 1. Ewolucja telefonów komórkowych

Kategoria	Telefony Analogowe	Telefony Cyfrowe	Smartphony
Kształt	„Cegła”	Płaskie, Składane, Rozsuwane	Płaskie, Składane, Rozsuwane
Waga	0,5 -1kg	180 – 250g	< 200g
Ekran	Niedostępny	Monochromatyczny lub kolorowy (max 256 kolorów), 170x120 pikseli	Kolorowy, 320x240 pikseli (lub więcej), 16 bitowa paleta kolorów
Procesor	Chip obsługujący komunikację głosową	Umożliwiający proste operacje	Stosunkowo wydajny procesor do zawansowanych zastosowań np. multimedia
Pamięć	Tylko do zapisu numerów telefonów	Do kilku megabajtów	64MB i więcej + wymienna pamięć flash do 4GB
Komunikacja (oprócz głosowej)	Niedostępna	Synchronizacja z komputerem stacjonarnym	Bluetooth, Wi-fi, GPS itp.
Bateria	Krótki czas gotowości i rozmów	Dłuższy czas rozmów (kilka godzin) i gotowości (kilka dni)	Dłuższy czas rozmów (kilka godzin) i gotowości (kilka dni)
Cena	Od kilkuset do kilku tysięcy dolarów	Darmowe (promocje) do paruset dolarów	Kilkaset dolarów lub mniej

Źródło: Zestawienie autora na podstawie: Pei Zheng, Lionel M. Ni , Smart Phone and Next-Generation Mobile Computing, 2006

Obecnie smartphony oprócz komunikacji zapewnianej przez technologie komórkową udostępniają po kilka metod bezprzewodowej komunikacji krótkiego zasięgu, takich jak Bluetooth, IrDa, Wi-fi. Smartphony umożliwiają mobilny dostęp do zewnętrznych źródeł danych, ale posiadają też znaczną pamięć lokalną. Potrafią przechowywać i przetwarzać dane na wiele różnych sposobów. Typowe smartphony posiadają wbudowaną kamerę wideo i umożliwiają zapis i odczyt audio/wideo. Standardowo zaopatrzone są w szereg aplikacji takich jak gry, komunikatory, klient poczty elektronicznej, przeglądarka internetowa i inne. Dodatkowo użytkownik nie jest ograniczony tylko do standardowego zestawu aplikacji. Systemy operacyjne zainstalowane na nowoczesnych smartphonach udostępniają środowiska i biblioteki programistyczne umożliwiające osobom niezwiązanym z producentem urządzenia lub autorem systemu operacyjnego na niemal dowolne tworzenie i uruchamianie nowych aplikacji. Dzięki temu smartphony mogę być wykorzystywane jako terminale dostępowe do usług handlu elektronicznego, systemów komputerowych, czy baz informacji[3].

W Tabeli 1. przedstawiono ogólne porównanie telefonów komórkowych trzech generacji. Warto zauważyć, że mimo spadku rozmiarów i wagi urządzeń ich moc obliczeniowa i zakres oferowanych funkcji nieustannie rośnie. Trend ten jest podobny do sytuacji obserwowanej w przypadku tradycyjnych systemów komputerowych. Można się spodziewać, że obecne smartphony w niedalekiej przyszłości będą uważane za „tradycyjne telefony komórkowe", a ich miejsce zajmie kolejna generacja smartphonów.

1.2 Sprzęt

Rozwój urządzeń mobilnych idzie w kierunku połączenia funkcjonalności sprzętowej i programowej telefonów komórkowych oraz urządzeń typu PDA i PocketPC. Smartphony przejmują również rolę mobilnych centrów rozrywki integrując możliwości odtwarzaczy MP3, przenośnych konsoli gier i małoformatowych odtwarzaczy wideo. To połączenie nie następuje jednak przez zwykłe dołączenie do telefonu komórkowego oddzielnej infrastruktury sprzętowej odpowiedzialnej za każdą z tych funkcji. Nową funkcjonalność uzyskuje się dzięki wykorzystaniu istniejącej uniwersalnej architektury sprzętowej.

1.2.1 Procesor

Istnieje wiele modeli i typów procesorów przeznaczonych do wykorzystania w urządzeniach mobilnych. Nawet popularne urządzenia takie jak odtwarzacze mp3, aparaty fotograficzne, czy telewizory posiadają wbudowane mikroprocesory. Jednak procesory wykorzystywane w urządzeniach mobilnych posiadają kilka charakterystycznych cech różniących je od mikroprocesorów wykorzystywanych w urządzeniach przemysłowych i tradycyjnych CPU wykorzystywanych w komputerach klasy PC:

- Ograniczona programowalność – mobilne procesory posiadają ograniczony zestaw instrukcji (w porównaniu z procesorami stacjonarnymi) ze względu na ograniczenia związane z poborem energii.

- Wysoką przepustowość operacji wejścia/wyjścia – zoptymalizowane instrukcje związane z wykorzystaniem sieci są częściej wykorzystywane na urządzeniach mobilnych.

- Optymalizacja przetwarzania strumieni danych – aplikacje multimedialne stanowią nieodzowną część urządzeń mobilnych, zatem procesory mobilne posiadają specjalne instrukcje dla obsługi tej funkcjonalności.

Najpopularniejsze typy procesorów mobilnych to:

ARM – (Advanced RISC Machine) Mikroprocesor opracowany przez brytyjską firmę ACRON z myślą o wykorzystaniu w systemach wbudowanych (ang. *embedded systems*). Obecnie najpopularniejsza architektura procesora wykorzystywana w smartphonach. W szczególności znany jest model StrongARM SA-100 wyprodukowany przez firmę DEC, oraz powstały na jego bazie intelowski XScale [15,18]. Procesory działają w trybie 32-bitowym. Zestaw instrukcji procesora ARM stanowi rozwinięcie zestawu instrukcji MOS 6502. Główne zmiany dotyczą zwiększenia efektywności potokowego przetwarzania instrukcji. Zgodnie z założeniami architektury RISC, rozkazy są tak skonstruowane, aby wykonywały się w ściśle określonym czasie – zwykle w jednym cyklu maszynowym. Unikatową cechą tego typu procesorów jest zdolność łączenia operacji na rejestrach z operacjami arytmetycznymi, dzięki czemu zmniejsza się liczba wymaganych operacji pobrania/zapisania argumentów, tym samym dodatkowo podnosząc efektywność potokowości[1].

MIPS – (Microprocessor without Interlocked Piped Stages) jest to architektura komputerowa (a w szczególności procesor typu RISC) rozwijana przez firmę MIPS Technologies. Istnieje zarówno w wersji 32-, jak i 64-bitowej. Procesory MIPS stanowią jednostkę centralną komputerów firmy SGI. Ponadto są szeroko stosowane w systemach wbudowanych, głównie w urządzeniach opartych na systemie operacyjnym Windows CE. Są używane w ruterach firmy Cisco oraz we współczesnych konsolach do gier, takich jak Nintendo 64, Sony PlayStation, Sony PlayStation 2, Sony PSP. Szacuje się, że procesory MIPS stanowią jedną trzecią produkcji mikroprocesorów typu RISC.

Crusoe i Efficeon – Procesory opracowane przez firmę Transmeta [19]. Wykorzystujące nowatorski system zarządzania energią (LongRun), umożliwiający dynamiczną i częstą zmianę napięcia i częstotliwości taktowania procesora w zależności od stopnia zapotrzebowania na moc obliczeniową zgłaszaną przez system operacyjny i uruchomione aplikacje. Korzystanie z technologii LongRun nie wymaga od systemu operacyjnego czy aplikacji żadnych specjalnych instrukcji. Crusoe i Efficeon są kompatybilne z procesorami typu x86 dzięki wykorzystaniu kodu pośredniego umożliwiającego tłumaczenie instrukcji procesorów x86 na natywne instrukcje VLIW wykorzystywane przez oba procesory.

1.2.2 Pamięć

Ograniczony zasób pamięci stanowi kolejne ograniczenie dla urządzeń mobilnych, wymuszając, aby system operacyjny i aplikacje mobilne zajmowały stosunkowo mało miejsca. Wyróżniamy trzy rodzaje pamięci: RAM, ROM i pamięć flash [4]. Obecne smartphony posiadają zwykle od 64MB do 128MB SRAM jako pamięć operacyjną dla aplikacji, około 128MB pamięci flash, na której przechowywamy jest kod systemu operacyjnego oraz od 128 do 256MB wbudowanej pamięci flash przeznaczonej na dane użytkownika. Większość smartphonów umożliwia wykorzystanie zewnętrznych wymienialnych kart pamięci takich jak SmartMedia, Compact Flash (CF), Memory Sticks i coraz popularniejsze Secure Digital (SD). Tego typu karty umożliwiają rozszerzenie pamięci na dane nawet o 4GB.

W odróżnieniu od tradycyjnych komputerów PC mobilne systemy operacyjny nie wykorzystują pamięci wirtualnej. Ze względu na stosunkowo mały rozmiar kodu systemu operacyjnego i aplikacji mobilnych cały wykonywany kod może być przechowywany w pamięci operacyjnej co znacząco poprawia wydajność.

Jako pamięć stałą w urządzeniach mobilnych wykorzystuje się przede wszystkim pamięć flash zamiast dysków twardych, ponieważ:

- umożliwia szybszy dostęp do danych niż dysk twardy,
- jest lżejsza i zajmuje znacznie mniej miejsca,
- jest cicha,

- nie zawiera wrażliwych na drgania części mechanicznych.

Z drugiej strony dyski twarde oferują znacznie większą pojemność niż pamięci flash. Również koszt wyprodukowania 1GB pamięci jest znacznie niższy w przypadku dysków twardych. Wraz z rozwojem multimediów dostępnych na telefony komórkowe będzie rosła potrzeba posiadania większej pamięci stałej i ewentualnie małe, wydajne i energooszczędne wersje dysków twardych mogą wyprzeć pamięci flash.

1.2.3 Urządzenia wejścia/wyjścia

Zestaw klawiszy i urządzeń wejścia/wyjścia jest bardzo istotny dla wygody korzystania z urządzenia mobilnego. Najpopularniejsze zestawy to:

- Numeryczna klawiatura telefoniczna – dwunasto przyciskowa klawiatura składająca się z cyfr od 0 do 9 (po wielokrotnym wciśnięciu umożliwiających również dostęp do liter) oraz klawisze * i #. Zwykle rozszerzona o 4 klawisze funkcyjne – rozmowy, kończenia rozmowy, menu i wstecz.

- Klawiatura *QWERTY* – miniaturowa wersja standardowej angielskiej klawiatury wykorzystywanej w stacjonarnych komputerach klasy PC. Dodatkowo niektóre klawisze podzielone są na 10 grup, tak aby wciśnięcie któregokolwiek z klawiszy w grupie symulowało wciśniecie cyfry jak w tradycyjnej klawiaturze telefonicznej.

- Ekrany dotykowe i klawiatury wirtualne – wykorzystywane głównie w urządzeniach PDA. Użytkownik przy pomocy specjalnego rysika (piórka) wskazuje elementy wyświetlone na ekranie. Urządzenia tego typu posiadają również zawansowane mechanizmy rozpoznawania pisma. Wadą tego rozwiązania jest to, że do obsługi urządzenia wymagane jest korzystanie z obu rąk.

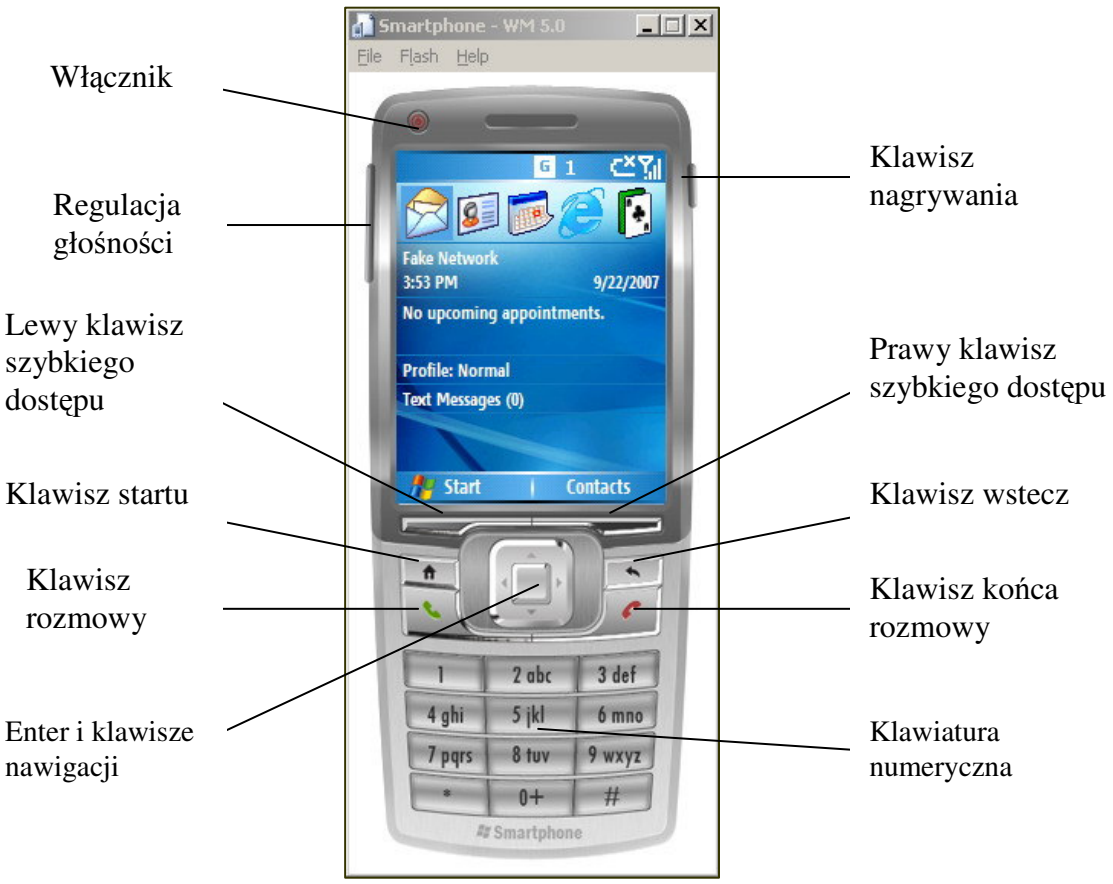
Najczęściej wykorzystywanym w smartphonach rozwiązaniem jest dwunasto przyciskowa klawiatura numeryczna zajmująca niewiele miejsca i wystarczająco poręczna. Wykorzystanie klawiatur QWERTY pozwala na sposób pisania bardziej zbliżony do korzystania z tradycyjnego komputera PC, ale niewielki rozmiar klawiszy znacząco utrudnia sprawne korzystanie. Oprócz standardowego zestawu klawiszy wiele modeli posiada dodatkowe klawisze funkcyjne. Na Rysunku 1. przedstawiono wygląd wzorcowego smartphona wraz z opisem dostępnych klawiszy.

1.2.4 Wyświetlacz

Ekrany telefonów komórkowych ewoluowały w wielu kierunkach w ciągu ostatnich kilku lat. Wczesne monochromatyczne wyświetlacze oferujące niską rozdzielczość zostały szybko wyparte przez kolorowe wyświetlacze wysokiej rozdzielczości. Rozmiary ekranów znacząco różnią się pomiędzy urządzeniami różnych producentów. Smartphony początkowo posiadały stosunkowo małe wyświetlacze, ale w związku ze stopniowym przejmowaniem roli PDA wymogiem staje się coraz większy ekran. Jednak wykorzystanie większych wyświetlaczy ujemnie wpływa na czas działania urządzenia na bateriach. Standardowe parametry ekranów to:

- **Rozmiar** – Telefony komórkowe mają zwykle ekrany LCD o przekątnej 2.2’ cale z opcją podświetlania i poprawy ostrości.

- **Rozdzielczość** – Zdarzają się jeszcze urządzenie z niską rozdzielczością (176 x 220 lub 128 x160), ale jako obecny standard przyjmuje się rozdzielczość QVGA (320x240) nowsze modele często posiadają już standard VGA (640x480) mimo wykorzystania tej samej przekątnej ekranu.



Rysunek 1.1: Standardowy zestaw klawiszy w Smartphonie [20]

- **Głębia kolorów** – najczęściej występujące modele posiadają 12-bitową (4096 kolorów) lub 16-bitową (65536 kolorów) głębię. Najbardziej zawansowane modele oferują 18-bitowy kolor (262144 kolory).

- **Konsumpcja energii** – Technologia Thin-film Transistor (TFT), w której wykonane są nowoczesne wyświetlacze potrzebuje znacznie więcej energii niż wykorzystywane w starszych modelach matryce pasywne. Typowy wyświetlacz w smartphonie zużywa około kilkuset miliwatów energii.

Ze względu na znaczący pobór mocy przez ekran i pozostałe elementy urządzenia niezbędne jest, aby mobilny system operacyjny oraz aplikacje udostępniały dodatkowe opcje zarządzania energią.

1.3 Systemy operacyjne

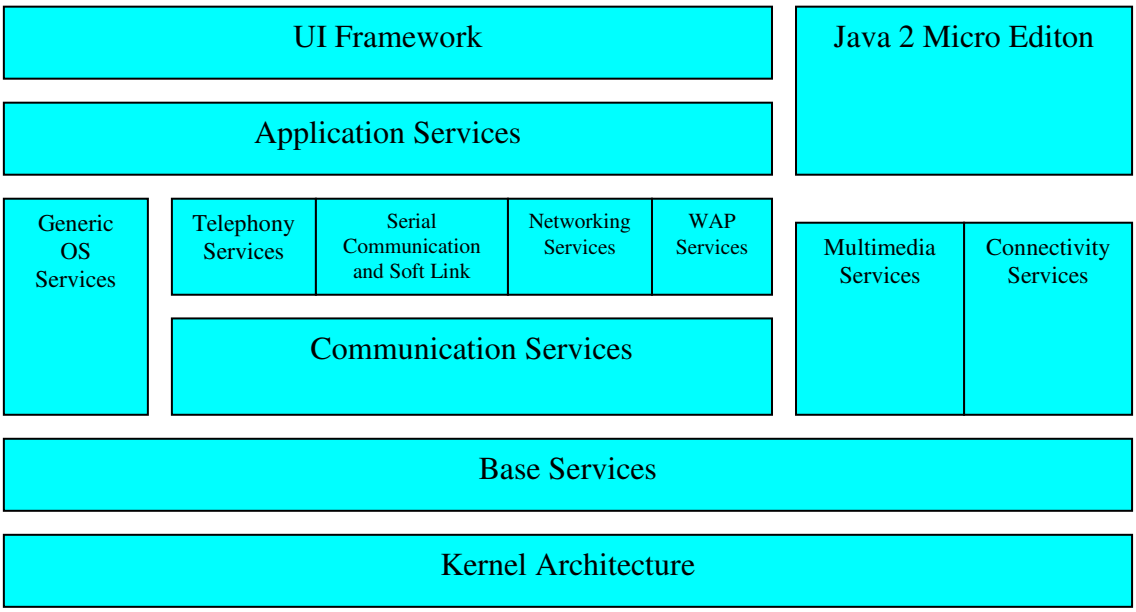
W odróżnieniu od tradycyjnych telefonów komórkowych smartphony posiadają wbudowany zaawansowany system operacyjny udostępniający, oprócz sterowników dla danego urządzenia, bogatą bibliotekę programistyczną, interfejs programowania aplikacji (API) oraz zestaw narzędzi umożliwiający niezależnym programistom tworzenie dodatkowych aplikacji na urządzenia mobilne. Pierwsze mobilne systemy operacyjny powstały z myślą o konkretnych modelach urządzeń, ale nowsze ich wersje można już dostosować do różnych typów urządzeń. Wraz z rozwojem technologii mobilnych coraz ważniejsze staje się, aby mobilny system operacyjny umożliwiał wygodny i efektywny sposób na tworzenie nowych aplikacji. W tym rozdziale przedstawione zostaną najpopularniejsze mobilne systemy operacyjne.

1.3.1 Symbian OS

System operacyjny Symbian został opracowany przez prywatną firmę o tej samej nazwie[16]. Współwłaścicielami firmy są wielcy producenci telefonów komórkowych, w tym Nokia, Sony Ericsson, Siemens i Samsung. Główną platformą, na którą przeznaczone są systemy Symbian są telefony komórkowe i smartphony. Ponad połowa sprzedawanych na świecie smartphonów działa właśnie pod kontrolą systemu Symbian.

Początkowo Symbian OS tworzony był na bazie systemu operacyjnego EPOC stworzonego przez firme Psion i wykorzystywanego w urządzeniach typu PDA. Najwcześniejsza wersja Symbiana to 6.0 kontynuując numerację systemów EPOC (ostatnia wersja to EPOC 5.0). Obecnie najnowsza komercyjna wersja Symbian OS ma numer 9.5. System operacyjny występuje w kilku odmianach. Wersja „Quarto" przeznaczona jest na urządzenia PDA, „Crystal" na tradycyjne telefony komórkowe, a „Series 60" na smartphony.

Symbian OS w wersji 9.5 umożliwia korzystanie z szerokiego wachlarza usług oferowanych przez systemy telefonii komórkowej 2G, 2.5G i 3G, jak również obsługę transmisji danych i multimediów. Oprócz standardowych usług umożliwia korzystanie z transmisji danych za pomocą protokołu TCP/IP, zawansowane funkcje kryptograficzne (3DES, RC5, AES, RSA) obsługę cyfrowych certyfikatów, szyfrowanie połączeń za pomocą TLS/SSL i WTLS przy korzystaniu z dostępu bezprzewodowego za pośrednictwem Wi-fi. Symbian OS działa na procesorach ARM lub symulujących procesory typu x86.



Rysunek 1.2: Architektura systemu operacyjnego Symbian 9.5 [16]

Symbian OS od wersji 8.0 posiada jądro wielowątkowe działające w czasie rzeczywistym, które zarządza pamięcią operacyjną, procesami i wątkami, zapewnia komunikację pomiędzy procesami, zarządzanie zasobami i obsługę wyjątków i błędów. Jądro działa natywnie wykorzystując instrukcje procesora ARM.

Usługi podstawowe (Base Services) udostępniają platformę programistyczną dla pozostały komponentów systemu. Umożliwiają one wywoływanie funkcji systemu operacyjnego, dostęp do sterowników urządzeń, systemu plików i standardowych bibliotek C++. Moduł usług komunikacyjnych (Communication Services) zapewnia dostęp do sieci, przesyłanie danych, wykorzystanie połączeń Bluetooth, IrDA i USB oraz programistyczny dostęp do standardowych

funkcji telefonu. Generic OS Services zapewnia typową funkcjonalność systemu operacyjnego (np. zarządzanie pamięcią, kontrola dostępu do plików). Moduł usług dla aplikacji (Application Services) umożliwia uruchomienia aplikacji użytkownika jako oddzielnych procesów, zapewnia mechanizm wyjątków i wsparcie dla różnych wersji językowych. Platforma interfejsu użytkownika (UI Framework) daje dostęp do standardowego zestawu kontrolek i zapewnia poprawną realizację mechanizmu zdarzeń.

Aplikację na Symbian OS pisze się wykorzystując zmodyfikowany język EPOC C++. Symbian umożliwia również uruchamianie aplikacji napisanych w Javie dla urządzeń mobilnych (J2ME) z wykorzystaniem mobilnej wersji maszyny wirtualnej Javy.

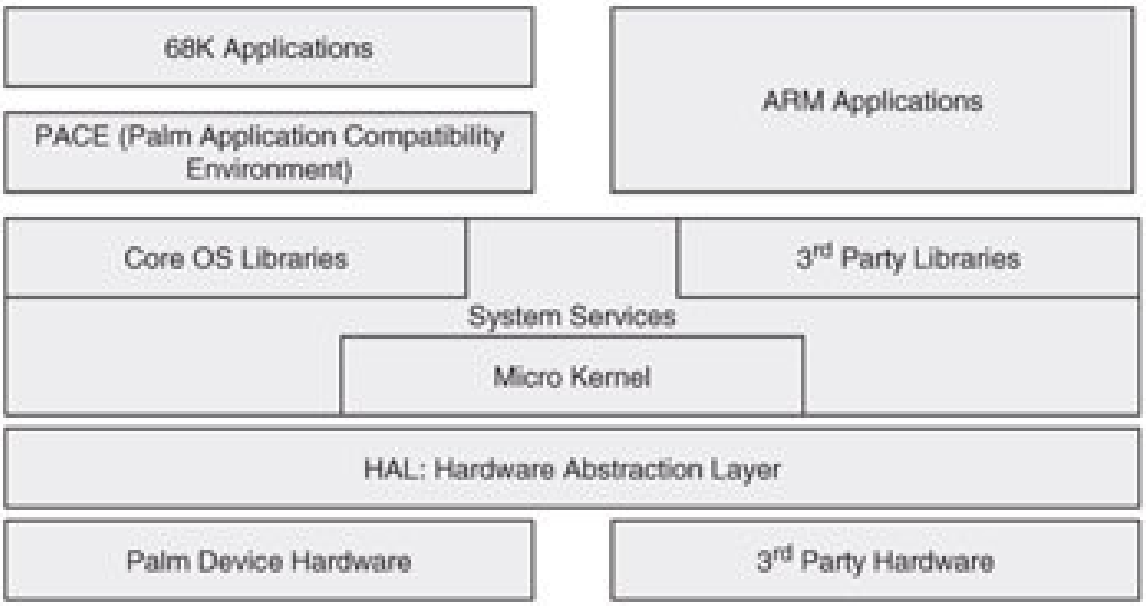
1.3.2 Palm OS

Firma Palm Inc [13] jest powszechnie znana jako wynalazca palmtopów, czyli pierwszych ogólnodostępnych i popularnych PDA. Palm OS powstał jako system operacyjny przeznaczony właśnie do Palm PDA. System operacyjny Palm wykorzystywany jest też w produktach Handspring, Sony i IBM. W grudniu 2006 r. firma Palm Inc. została przejęta przez japońską firmę ACCESS, która na bazie systemu Palm opracowuje dwa nowe mobilne systemy operacyjne ACCESS Linux Platform i Garnet OS [8].

Do wersji Palm OS 4.0 twórcy systemu jako główny cel przyjmowali stworzenie systemu, który byłby maksymalnie efektywny przy użyciu jak najmniejszej mocy obliczeniowej. Aplikacje na Palm OS były przede wszystkim nakierowane na interakcję z człowiekiem i posiadały bardzo przyjazny interfejs użytkownika. Największą wadą pierwszych wersji był brak możliwości uruchamiania kilku zadań jednocześnie, co powodowało, że w danym momencie mogła działać tylko jedna aplikacja. Nie miało to większego znaczenia, gdyż liczba dostępnych aplikacji była stosunkowo niewielka. W zamian system oferował znakomitą wydajność z punktu widzenia użytkownika – brak jakichkolwiek opóźnień podczas korzystania z aplikacji.

Wraz z popularyzacją urządzeń PDA i rosnącymi wymaganiami klientów co do możliwości zainstalowanych aplikacji, niezbędne stało się znaczące unowocześnienie systemu operacyjnego. W nowych wersjach systemu Palm OS 5.0 i 6.0 wprowadzono znaczącą liczbę ulepszeń. Od wersji 5.0 Palm OS może być instalowany na procesorach typu ARM (wcześniej tylko na procesorach Motoroli z serii 68K), ale udało się zachować kompatybilność wstecz i aplikacje napisane na starsze wersje systemu bez problemu uruchamiały się w wersji 5.0. Wprowadzono wielozadaniowość, obsługę LAN, Bluetooth, usługi telefonii komórkowej (od wersji 5.0 możliwe jest wykorzystanie Palm OS w smartphonach) i odświeżono interfejs użytkownika.

Prawdziwa rewolucją nastąpiła jednak wraz z wejściem wersji 6.0, która została napisana niemal od zera z myślą o przygotowaniu uniwersalnego mobilnego systemu operacyjnego. Palm OS 6.0 można łatwo dostosować do różnych architektur sprzętowych dzięki zastosowaniu warstwy pośredniej pomiędzy sprzętem a systemem operacyjnym (Hardware Abstraction Layer). Nowa wersja systemu obsługuje wielowątkowość, zapewnia zarządzanie pamięcią obsługę zdarzeń GUI, oraz wsparcie dla multimediów. Dołączono też pełną obsługę popularnych metod komunikacji bezprzewodowej (Wi-fi, Bluetooth). Moduł Core OS obsługuje niskopoziomowy dostęp do systemu plików oraz obsługę protokołu TCP/IP. Ciekawym dodatkiem jest możliwość rozszerzania systemu o biblioteki napisane przez firmy zewnętrzne, tym samym znacząco zwiększając wydajność ich wykonywania. (np. obsługa J2ME została tak wbudowana w system). Warstwa „Palm Application Compatibility Environment” umożliwia uruchomienie aplikacji napisanych na starsze wersje systemu. W systemie operacyjnym Palm wykorzystano niestandardowy model systemu plików. Zamiast trzymać pliki blokowo zapisane w pamięci Palm OS wykorzystuje wewnętrzną mikro bazę danych (Palm DataBase).



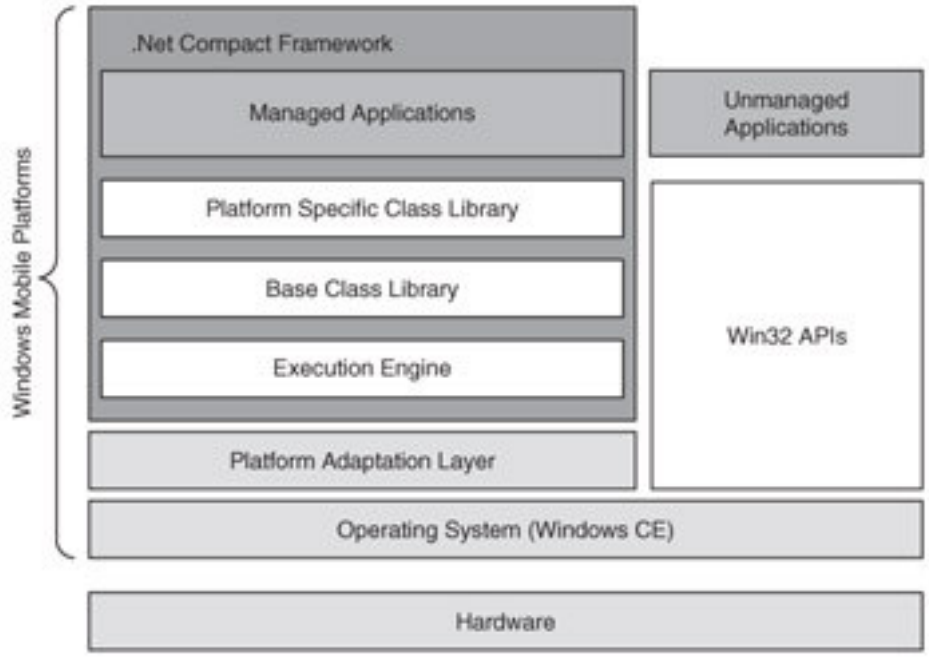
Rysunek 1.3: Architektura Palm OS 6.0 [13]

Przy pisaniu aplikacji na Palm OS wykorzystuje się przygotowany przez producenta Palm OS 68K/Protein SDK. Programiści mogą korzystać z języków C, C++, Visual Basic i Javy, ale większość aplikacji dla Palm OS powstaje w C. SDK zawiera kompilatory pozwalające na tworzenie aplikacji działających natywnie na procesorach ARM dla Palm OS 6.0, jak również przeznaczone dla starszych wersji systemu wykorzystujących procesory 68K. Jako środowisko programistyczne wykorzystuje się Metrowerks CodeWarrior, Palm PRC Tools lub Eclipse.

1.3.3 Windows Mobile

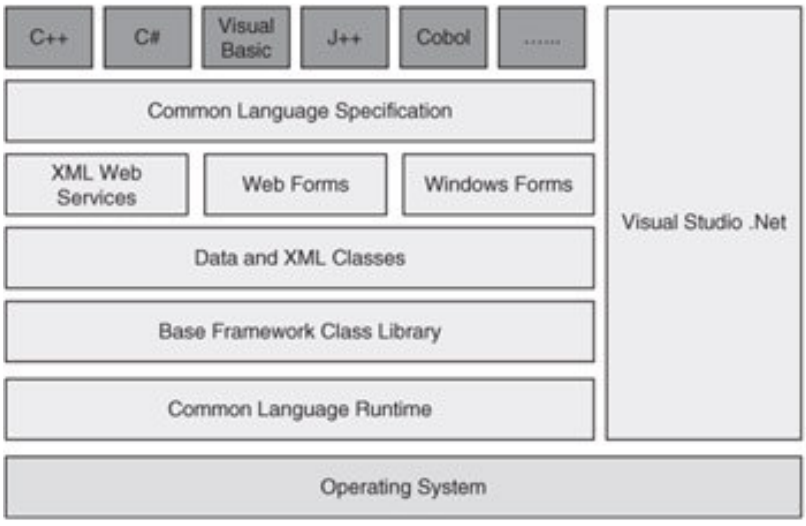
System operacyjny Windows Mobile [23] został opracowany przez firmę Microsoft na podstawie systemu Windows CE, który zaprojektowany był z myślą o niewielkich urządzeniach mobilnych głównie PDA, telefonach komórkowych, naręcznych komputerach i systemach nawigacji satelitarnej. Środowisko Windows Mobile występuje w dwóch wersjach, dla PocketPC (PDA korzystające z oprogramowania Microsoft) oraz SmartPhone. Obie wersje korzystają z Windows CE, a różnice między nimi są stosunkowo niewielkie i związane z innym sposobem komunikacji użytkownika z urządzeniem (PDA – rysik, smartphone – klawiatura). Większość aplikacji napisanych dla smartphona można z powodzeniem wykorzystywać na PDA. Wraz z przejmowaniem przez smartphony roli PDA można się spodziewać całkowitego połączenia obu wersji. Najnowsza wersja Windows Mobile ma numer 6.0.

Środowisko .NET Compact Framework [5] wraz z dołączonymi do niego bibliotekami programistycznymi umożliwia pisanie aplikacji w kodzie zarządzanym. Oprócz tego wpierane jest też tradycyjne Win32 API oraz Microsoft Foundation Classes (MFC) umożliwiające tworzenie aplikacji uruchamianych w kodzie natywnym. Na rysunku 1.4 przedstawiono ogólną architekturę systemu.



Rysunek 1.4: Windows Mobile [21]

Głównym elementem platformy Windows Mobile jest .Net Compact Framework, który ładowany jest do pamięci ROM smartphona. Compact Framework jest mobilną wersją środowiska .NET Framework dostępnego dla stacjonarnych komputerów PC. Ogólnie rzecz biorąc platforma .NET powstała z myślą o wygodnym, bezpiecznym i uniwersalnym tworzeniu aplikacji wykorzystujących szeroki zakres usług Internetowych. Środowisko .NET Framework intensywnie wykorzystuje eXtensible Markup Language (XML) i usługi sieciowe (ang. *web services*) do wymiany informacji pomiędzy różnymi niejednolitymi systemami działającymi w sieci Internet. Jest to możliwe dzięki zastosowaniu otwartego standardu XML pozwalającego na zintegrowanie przesyłania danych i ich struktury w samoopisującym się formacie w taki sposób, że mogą być one przetwarzane, modyfikowane i wymieniane pomiędzy dowolnymi aplikacjami, serwisami internetowymi, czy innymi urządzeniami mobilnymi. Mechanizm usług sieciowych wykorzystuje oparty na XML protokół SOAP (Simple Object Access Protocol) do transportu danych pomiędzy aplikacją a serwisem oraz standard UDDI (Universal Data Description Interface) do wyszukiwania i katalogowania usług sieciowych. Na rysunku 1.5 przedstawiono schemat platformy .NET.



Rysunek 1.5: Microsoft .Net Framework [11]

Infrastruktura .NET Framework jest podobna dla wszystkich urządzeń wykorzystujących systemy Windows (stacjonarne PC, serwery, tablety PC, Pocket PC i Smartphony) i składa się z dwóch głównych komponentów: wirtualnej maszyny (Common Language Runtime – CLR) i biblioteki klas. CLR jest warstwą pośredniczącą pomiędzy uruchamianym kodem aplikacji a systemem operacyjnym. Do pisania aplikacji programiści mogą korzystać z bogatego zestawu języków programowania – C, C++, C#, Visual Basic, Fortran, Perl. Niezależnie od wybranego języka programowania kod aplikacji kompilowany jest do języka pośredniego (Intermediate Language – IL lub MSIL). Kod pośredni uruchamiany jest przy użyciu CLR, który kompiluje go w locie do kodu natywnego maszyny, na której jest uruchomiony. Platforma .NET zapewnia zarządzanie pamięcią (w tym odśmiecanie), kontrolę typów (Common Type System – CTS), obsługę wyjątków i kontrolę wykonywanego kodu. Biblioteka klas zawiera zestaw klas i interfejsów dostępnych dla wszystkich zgodnych z .NET języków programowania. Zawartość biblioteki można podzielić na 3 grupy: ASP.NET (dla stron i aplikacji webowych), ADO.NET (klasy odpowiedzialne za dostęp i przetwarzanie danych) i Windows Forms (kontrolki i klasy do tworzenia interfejsu użytkownika).

Dla urządzeń mobilnych przygotowana została bardzo okrojona wersja .NET Framework – .NET Compact Framework [5]. Podobnie jak pełna wersja składa się z dwóch komponentów: CLR i biblioteki klas. Ze względu na ograniczone możliwości przetwarzania i niewielką pojemność pamięci urządzeń mobilnych Compact Framework CLR jest o wiele mniejszy (zajmuje mniej niż 2 MB) i znacznie wydajniejszy niż CLR na systemy stacjonarne. Biblioteka klas zawiera tylko podzbiór klas oryginalnej biblioteki, które zostały dodatkowo zoptymalizowane (lub napisane na nowo) pod urządzenia mobilne. Biblioteka zawiera też nowe klasy do wykorzystywania usług i tworzenia aplikacji specyficznych dla urządzeń mobilnych.

Jako środowisko do tworzenia aplikacji na Windows Mobile można wykorzystać flagowe narzędzie Microsoftu – Visual Studio .NET, które udostępnia specjalne projekty dla aplikacji na PocketPC, smartphone i Windows CE. Programiści mogą wybrać język C#, C++ lub Visual Basic dla aplikacji w kodzie zarządzanym lub C++ dla aplikacji korzystających z Win32 API. Compact Framework jest dynamicznie rozwijany i regularnie dodawane są nowe wersje i funkcjonalności. Jeśli jakaś funkcjonalność nie jest wspierana przez CLR programista może wywołać tradycyjną natywną funkcję z Windows CE API, omijając tym samym maszynę wirtualną (tak zwany Platform Invoke lub P/Invoke). Aby rozpocząć programowanie aplikacji dla Windows Mobile należy zaopatrzyć się w zestaw kompilatorów i narzędzi programistycznych Windows Mobile SDK [22] w wersji dla wybranego urządzenia mobilnego (PocketPC lub smartphone). Podczas fazy implementacji nie jest wymagane korzystanie z rzeczywistego urządzenia. W celach deweloperskich i testowych można wykorzystać emulator urządzenia mobilnego. Emulator nie tylko symuluje wygląd i funkcjonalność rzeczywistego urządzenia, ale również emuluje działanie sprzętu (procesora ARM, usług telefonicznych itp.). Zaleca się jednak systematyczne sprawdzanie działania aplikacji na urządzeniu docelowym [11].

1.3.4 Embedded Linux

Linux jest darmowym systemem operacyjnym, o ogólnie dostępnym otwartym kodzie źródłowym. Jądro Linuksa utrzymywane jest i rozwijane przez niezależnych programistów kierowanych przez Linusa Torvalds – twórcę systemu operacyjnego. Istnieje znaczna liczba darmowych aplikacji napisanych dla systemu Linux. Kody źródłowe tych aplikacji, jak również samego jądra, dostępne są na zasadach licencji GPL, która mówi, że każdy wykorzystujący w swojej aplikacji kod objęty licencją GPL musi zapewnić swobodny dostęp do tworzonego przez siebie programu zgodnie z licencją GPL. System operacyjny Linux jest uznawany za rozsądną alternatywę dla komercyjnych systemów operacyjnych [7].

Próba przeniesienia Linuksa na platformę urządzeń mobilnych podejmowana jest przez kilka komercyjnych firm, jak również niezależnych programistów. Dystrybucje Linuksa na PDA są dostępne do ściągnięcia z Internetu. Istnieją też komercyjne wersje (np. Monta Vista Linux),

szczególnie popularne wśród producentów sprzętu z Azji. Nad wersją Linuksa na smartphony pracują takie firmy jak Motorola i NEC.

Poszczególne dystrybucje bardzo różnią się od siebie, ale można wskazać szereg wspólnych cech:

- Jednolite jądro systemu wspierające wielozadaniowość i wielowątkowość, które może być dodatkowo prekonfigurowane dla urządzenia i działających aplikacji.
- Dzięki aktywnej społeczności programistów dostępne są sterowniki i rozwiązania dla najnowszych technologii komunikacyjnych (w tym Bluetooth, Wi-fi, LAN itp.).
- Aplikacje dystrybuowane na licencji GPL są łatwo modyfikowalne i rozszerzalne.
- Opłata licencyjna jest niewielka (albo w ogóle nie istnieje).

Linux może być stosunkowo łatwo przystosowywany do różnych urządzeń mobilnych, w związku z tym bardzo ważne staje się wypracowanie wspólnych dla różnych dystrybucji standardów, aby poszczególne dostosowane do urządzenia dystrybucje zachowały wzajemną zgodność i umożliwiały uruchomienie wspólnego zestawu aplikacji. Opracowywaniem standardów zajmuje się dwie organizacje Consumer Electronics Linux Forum i Embededded Linux Consortium [6].

1.3.5 Java 2 Mobile Eddition (J2ME)

Java ME [10] nie jest systemem operacyjnym, ale właściwe mobilne systemy operacyjne często wykorzystują Javę dla zapewnienia przenośności kodu i wykorzystania aplikacji, które nie zostały napisane dla konkretnego urządzenia. Aplikacje napisane w Javie kompilowane są do bajtkodu (języka pośredniego), który następnie jest uruchamiany na wirtualnej maszynie Javy opracowanej dla konkretnego mobilnego systemu operacyjnego. Java działa tu jako warstwa pośrednicząca pomiędzy specyficznym mobilnym systemem operacyjnym a aplikacjami oferowanymi przez niezależnych producentów.

Istnieją dwa rodzaje konfiguracji J2ME w zależności od możliwości urządzenia, na którym jest instalowana

- CLDC – dla urządzeń z procesorami 16 lub 32-bitowymi o zegarze minimum 16Mhz. Oferujących około 512KB pamięci i ograniczony dostęp do sieci. Do tej grupy należą starsze modele telefonów komórkowych i PDA, pagery, odtwarzacze muzyczne itp.
- LDC – dla urządzeń z wydajnymi 32-bitowych procesorami, udostępniających przynajmniej 2MB RAM pamięci operacyjnej i 2,5MB ROM dla instalacji wirtualnej maszyny. Przykłady urządzeń to nowoczesne PDA i Smartphony

Mimo zastosowania wirtualnej maszyny Javy zdarza się, że aplikacje nie uruchamiają się poprawnie na różnych urządzeniach. Jest to spowodowane tym, że niektóre aplikacje wymagają zastosowania wywołań systemu operacyjnego nieudostępnianych przez maszynę wirtualną dla konkretnego urządzenia.

Obecnie J2ME jest szeroko stosowana do tworzenia aplikacji na urządzenia mobilne, zarówno gier, jaki i rozwiązań biznesowych.

1.4 Aplikacje i usługi

Zestaw aplikacji i usług dostępnych na smartphony stale rośnie. Telefon komórkowy przejmuje rolę PDA i mobilnego centrum rozrywki, co skutkuje dostępem do nowej funkcjonalności. Nowe aplikacje i usługi powstają również wraz z rosnącą przepustowością łącz, upowszechnianiem się bezprzewodowych publicznych sieci bezprzewodowych i powstaniem nowych modeli

biznesowych wykorzystujących technologie mobilne. Obecnie istniejące aplikacje możemy podzielić na dwie generacje, opisane w kolejnych punktach [14].

1.4.1 Aplikacje pierwszej generacji

Aplikacje należące do pierwszej generacji to głównie naturalne rozszerzenia funkcjonalności telefonów komórkowych lub PDA. Większość z nich jest szeroko dostępna i wykorzystywana na co dzień. Do aplikacji pierwszej generacji należą:

- Bezprzewodowa komunikacja głosowa – tradycyjna usługa telefonii komórkowej pozwalająca na odbywanie rozmów głosowych oraz korzystanie z poczty głosowej. Usługa ta jest i najprawdopodobniej pozostanie najważniejszym aspektem wykorzystania telefonów komórkowych.

- Krótkie wiadomości tekstowe (SMS) – usługa umożliwia wysyłanie oraz otrzymywanie informacji tekstowych pomiędzy użytkownikami telefonów komórkowych.

- Multimedialne wiadomości tekstowe (MMS) – umożliwiają przesyłanie sformatowanego tekstu, animacji, zdjęć, melodii, a nawet sekwencji wideo na podobnych zasadach jak SMS.

- System nawigacji satelitarnej (GPS) – usługa umożliwiająca śledzenie położenia telefonu komórkowego nawet w skali świata. Zakres wykorzystanie tej funkcjonalności jest określany przez prawo poszczególnych krajów.

- System informacji geograficznej (GIS) – pozycja telefonu komórkowego określana jest z wykorzystaniem systemu GPS, a następnie odpowiednie do położenia informacje przekazywane są do aplikacji na telefonie. Informacje te obejmują mapy, zalecaną trasę do celu, informacje o przeszkodach itp.

- Komunikatory internetowe – usługa pozwalająca na prowadzenie rozmów w czasie rzeczywistym z wykorzystaniem wiadomości tekstowych. Wiele znanych z komputerów PC komunikatorów posiada swoje wersje na telefony komórkowe.

- Poczta elektroniczna, kalendarz, notatnik, książka adresowa itp. – typowe aplikacje do zarządzania osobistymi informacjami i kontaktami. Większość tych aplikacji umożliwia synchronizację danych z ich odpowiednikami na komputery stacjonarne.

- Bezprzewodowy dostęp do stron WWW – dostęp do sieci Web stanowi podstawową funkcjonalność. Przy użyciu smartphona można łączyć się ze stronami specjalnie przygotowanymi z myślą o użytkownikach mobilnych, jak również z tradycyjnymi stronami WWW.

- Dostęp do Internetu – połączenie Internetowe zainicjowane przez telefon komórkowy może być następnie wykorzystane przez inne urządzenie np. laptop.

- Serwisy tematyczne – wiadomości, pogoda, informacje finansowe, blogi itp. Dane mogą być dostarczane w określonych odstępach czasu bezpośrednio do telefonu komórkowego, a następnie przeglądane przy użyciu specjalnych programów.

- Strumienie audio/wideo – bezpośredni dostęp do plików muzycznych i filmowych. Usługi dostępne tylko w sieciach komórkowych wykorzystujących technologie 2.5G lub 3G (UMTS).

1.4.2 Aplikacje drugiej generacji

Gwałtowny rozwój komunikacji bezprzewodowej skutkuje powstaniem szerokiego wachlarza nowych usług wymagających znacznie większej przepustowości łącza. Niektóre z wymienionych aplikacji są już wykorzystywane w niektórych krajach, inne pozostają w fazie rozwoju.

- Voice over IP (VoIP) – technologia umożliwiająca przesyłanie dźwięków mowy za pomocą łączy internetowych lub dedykowanych sieci, popularnie nazywana telefonią internetową. Dane przesyłane są przy użyciu protokołu IP, co pozwala wykluczyć niepotrzebne "połączenie ciągłe" i np. wymianę informacji, gdy rozmówcy milczą. W przypadku smartphonów wykorzystywane byłyby lokalne sieci WLAN.

- Mobilny handel elektroniczny – obejmuje zakupy online, operacje giełdowe, reklamę, informacje na temat pobliskich punków usługowych. Nie wszystkie aplikacje z tej grupy skupiają się na sprzedaży dóbr, często o wiele ważniejsze jest informowanie klienta o zmieniających się ofertach.

- Rozwiązania biznesowe – Firmy zatrudniające pracowników, którzy pozostają w ciągłym ruchu mogą wykorzystać smartphony do ułatwienia komunikacji i współpracy pomiędzy pracownikami. Technologie mobilne są obecnie wykorzystywane przez firmy spedycyjne, policję, przedstawicieli handlowych, taksówkarzy itp. Do tej pory mogli oni odbierać nowe przydziały czy zlecenia tylko w biurze firmy lub dzwoniąc do centrali. Dzięki wykorzystaniu urządzeń mobilnych mają ciągły i natychmiastowy dostęp do najświeższych informacji i na tej podstawie mogą efektywnie zarządzać realizacją przydzielonych im zadań.

- Mobilne gry multiplayer – wraz z popularyzacją telefonii komórkowej 3G spodziewany jest wzrost popularności internetowych gier multiplayer dostępnych z telefonów komórkowych. Pierwsze tego typu gry są już dostępne nawet w Polsce (np. TibiaME). Obecnie gry multiplayer napisane na smartphony do rozgrywek wieloosobowych wykorzystują bezprzewodowe sieci typu *ad hoc*. Główni producenci konsol do gier – Nintendo i Sony zapowiedzieli wypuszczenie na rynek własnych urządzeń mobilnych klasy smartphone specjalnie przystosowanych do mobilnej rozrywki.

- Serwisy muzyczne – po sukcesie serwisów internetowych oferujących odpłatne ściąganie utworów muzycznych autorzy serwisów w porozumieniu z operatorami telefonii komórkowej przygotowują podobną usługę dla użytkowników urządzeń mobilnych. Tym bardziej, że wielu posiadaczy smartphonów wykorzystuje je jako przenośne odtwarzacze MP3.

- Zdalny dostęp do komputerów stacjonarnych – za pośrednictwem telefonu komórkowego będzie możliwe połączenie się i zarządzanie domowym komputerem stacjonarnym przy wykorzystaniu prywatnych wirtualnych sieci (VPN).

- Mobilny portfel i bilet – smartphone z zainstalowanym odpowiednim oprogramowaniem będzie mógł być wykorzystany jako elektroniczna portmonetka. Obecnie w wielu krajach (w tym w niektórych miastach w Polsce, np. w Kielcach) można kupować bilety komunikacji miejskiej za pośrednictwem SMS. Specjalny kod otrzymany w zwrotnym SMS stanowi dowód zakupu biletu.

- Społeczności internetowe – specjalne aplikacje umożliwiające wygodne korzystanie z portali internetowych społeczności.

- Mobilny identyfikator – specjalne aplikacje zainstalowane na smartphonie mogą być wykorzystywane jako elektroniczny identyfikator, legitymacja studencka, indeks, prawo jazdy, klucz do domu, karta biblioteczna itp. Niektóre urządzenia mobilne oprócz

tradycyjnych zabezpieczeń w postaci identyfikatora i hasła oferują czytniki linii papilarnych.

2. Wizja systemu i wymagania

2.1 Wizja

Otaczający nas świat ciągle się zmienia, atakuje nas natłok informacji wymagających od człowieka umiejętności wybrania tych istotnych i wartych zapamiętania. Rosnące tempo życia zmusza nas do skupienia się nad wieloma sprawami jednocześnie, co nie zawsze jest możliwe i sprawia, że zapominamy o niektórych rutynowych codziennych czynnościach. Nowoczesne technologie mobilne umożliwiają nam lepsze wykorzystanie naszego czasu i pamiętają za nas o tych „drobnych codziennych sprawach".

Wielu zabieganych i zapracowanych ludzi nie ma czasu robić codziennych zakupów. Wielokrotnie zdarza się, że czegoś brakuje w domu, a nawet jak znajdzie się chwila wolnego czasu (np. po drodze z pracy do domu), to nie pamiętamy, co tak naprawdę mamy kupić. Tradycyjnym sposobem radzenia sobie z tym problemem jest zapisywanie produktów, które mamy zamiar nabyć przy następnej wizycie w sklepie, na specjalnie do tego przygotowanej papierowej kartce umieszczonej najczęściej na drzwiach lodówki. Jednak takie rozwiązanie ma zasadniczą wadę. W wielu przypadkach, kiedy uda nam się znaleźć czas na zakupy nie mamy tej kartki przy sobie i zdajemy się na naszą pamięć. Po pewnym czasie rezygnujemy z zapisywania, bo jak przyjdzie czas zakupów i tak będziemy musieli pamiętać całą listę.

Sam pomysł z zapisywaniem listy zakupów nie jest zły, a wady tego rozwiązania związane są z zastosowaniem „nie mobilnego" nośnika, jakim jest kartka papieru. Lista zakupów powinna znajdować się na nośniku, który zawsze mamy przy sobie. Doskonałym do tego zadania wydaje się telefon komórkowy, który stał się już obowiązkowym codziennym wyposażeniem nowoczesnego człowieka. Głównym zadaniem projektowanego systemu działającego na nowoczesnych telefonach komórkowych (smartphone), będzie zarządzanie listą zakupów dla pojedynczego użytkownika i całej rodziny.

2.2 Wymagania funkcjonalne

Zarządzanie listą zakupów

Podstawowym zadaniem systemu jest oczywiście zarządzanie listą zakupów. Użytkownik musi mieć możliwość tworzenia, edytowania i usuwania pojedynczej listy i wielu niezależnych list. Istotne jest, aby przy tworzeniu lub edycji listy dać użytkownikowi sposobność wyboru produktów z wcześniej przygotowanej bazy artykułów, jak również możliwość rozszerzenie tej bazy o nowe produkty. Lista zakupów powinna być prezentowana w sposób przejrzysty oraz umożliwiać zaznaczenie już zakupionych rzeczy.

Przesyłanie listy innym użytkownikom

System powinien zapewniać możliwość przesyłania stworzonej listy zakupów innym użytkownikom systemu (np. członkom rodziny). Często dochodzi do sytuacji, że chcielibyśmy poprosić wracającego ze szkoły lub pracy członka rodziny, aby kupił po drodze wybrane artykuły. Zamiast dzwonić do niego możemy po prostu wysłać mu wcześniej przygotowaną listę zakupów. Istotne jest, aby użytkownik otrzymujący taką listę został o tym powiadomiony natychmiast po jej otrzymaniu.

Wymiana list

Każdy z użytkowników systemu powinien mieć możliwość dodawania produktów do listy zakupów znajdującej się na jego osobistym telefonie. System powinien zapewniać wymianę i łączenie tych list pomiędzy użytkownikami systemu dokonującymi wspólnych zakupów, ale korzystających z fizycznie różnych urządzeń.

Rezerwacja produktów

Dodatkową funkcją systemu będzie możliwość zamówienia wybranych przez nas artykułów w pobliskim sklepie wielobranżowym. Zamówione artykuły będzie można odebrać osobiście lub zamówić dostawę do domu.

Prostota i szybkość korzystania z aplikacji

Bardzo ważne jest, aby korzystanie z systemu było jak najprostsze i jak najszybsze. Czas potrzebny na dodanie nowego artykułu do listy zakupów nie powinien być znacząco dłuższy niż czas potrzebny na zapisanie tej informacji na kartce papieru.

Możliwość konfigurowania aplikacji mobilnej za pomocą standardowego komputera PC

Operacje, które wymagają większego nakładu pracy od użytkownika (np. dodanie dużej ilości nowych artykułów do bazy produktów) powinny się odbywać po podłączeniu urządzenia mobilnego do tradycyjnego komputera PC, aby mógł on skorzystać z wygodnych urządzeń peryferyjnych, jak pełnowymiarowa klawiatura i mysz.

2.3 Wymagania niefunkcjonalne

Zapewnienie komunikacji pomiędzy aplikacjami bez udziału zewnętrznych serwerów

Aplikacja działająca na telefonie komórkowym powinna działać samodzielnie niezależnie od zewnętrznych serwerów, chyba że wykorzystywana funkcjonalność wymaga kontaktu z serwerem zewnętrznym (np. przy zamawianiu produktów). Aplikacje na różnych fizycznie aparatach telefonicznych powinny porozumiewać się bezpośrednio.

Możliwość wykorzystania aplikacji na urządzeniach typu smartphone, jak również PDA-Phone

Wskazane jest takie zaimplementowanie systemu, aby można z niego było korzystać na różnych popularnych typach urządzeń takich jak smartphone i PDA. Oczywiście niektóre mniej istotne funkcje systemu mogą być dostępne tylko na konkretnym urządzeniu (np. wykorzystanie rysika, gdy aplikacja uruchamiana jest na PDA)

Możliwość dodawania nowych produktów i kojarzenia artykułów z listy z dostępnymi w sklepie

Przy korzystaniu z funkcji zamawiania produktów w pobliskim sklepie istotne jest, aby system sam potrafił dobrać produkty dostępne w asortymencie sklepu do zamawianych przez klienta (np. w sklepie może znajdować się wiele typów „mleka", które znalazło się na liście zakupów klienta). System powinien dawać klientowi sposobność wyboru konkretnego produktu z listy sugerowanych, jak również możliwość automatycznego doboru.

Umożliwienie prostego uaktualnienia aplikacji mobilnej

Ważne jest, aby system udostępniał mechanizm prostego i w miarę możliwości jednoczesnego uaktualnienia programu na kilku urządzeniach mobilnych. Uaktualnienia mogą obejmować nową wersję samej aplikacji mobilnej, jak również uaktualnioną bazę danych produktów.

Wydajność i skalowalność

Ze względu na stosunkowo małą wydajność urządzeń mobilnych aplikacja powinna wykonywać minimalną wymaganą ilość obliczeń. Jeżeli istnieje taka możliwość, to obliczenia powinny być wykonywane przez serwery zewnętrzne lub podczas konfiguracji aplikacji, gdy urządzenie mobilne podłączone jest do tradycyjnego komputera klasy PC.

Niedopuszczalne jest, aby brak możliwości skontaktowania się z serwerem zewnętrznym uniemożliwiał poprawne działanie głównych funkcji aplikacji.

Niewielka rozmiar pamięci urządzenia mobilnego wymaga ciągłego dostosowywania przechowywanej bazy produktów, aby była ona jak najmniejsza, a jednocześnie w pełni funkcjonalna.

Minimalna komunikacja pomiędzy aplikacjami mobilnymi

Łączność pomiędzy urządzeniami mobilnymi odbywa się za pomocą odpłatnych usług oferowanych przez operatora telekomunikacyjnego (SMS, GPRS). Komunikacja pomiędzy aplikacjami mobilnymi powinna być zredukowana do minimum, aby obniżyć finansowe koszty jej wykorzystania.

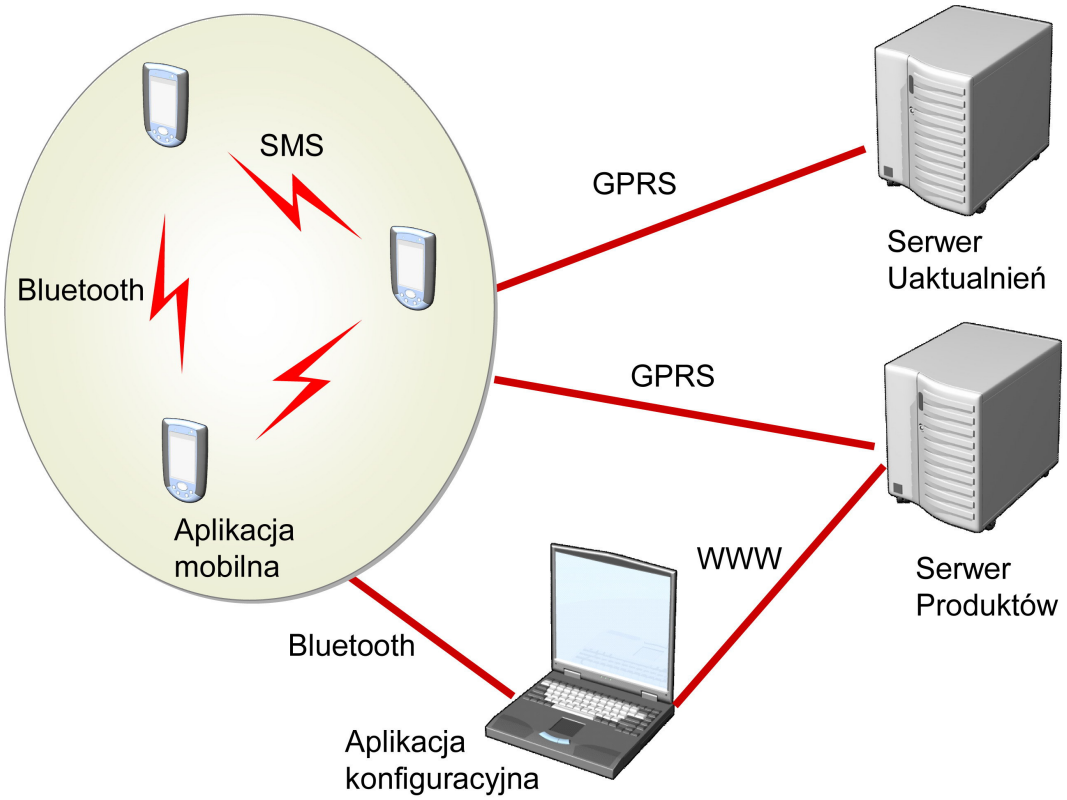
3. Projekt systemu

3.1 Ogólna architektura systemu

Na system składają się cztery w dużym stopniu niezależne programy: aplikacja mobilna, serwer produktów, aplikacja konfiguracyjna i serwis uaktualnień.

Głównym elementem systemu jest aplikacja mobilna uruchamiana na jednym lub wielu urządzeniach mobilnych typu smartphone lub PDA. Aplikacja mobilna przez większość czasu działa niezależnie od pozostałych elementów systemu. Łączność pomiędzy aplikacją mobilną, a serwerem produktów lub serwerem uaktualnień odbywa się za pomocą połączenia w technologii General Packet Radio Service (GPRS) i Wireless Application Protocol (WAP) w tradycyjnej architekturze klient-serwer. Poszczególne instancje aplikacji mobilnej na różnych urządzeniach fizycznych porozumiewają się niezależnie od pozostałych elementów systemu zgodnie z modelem punkt-do-punktu. Komunikacja odbywa się za pomocą krótkich wiadomości tekstowych (SMS) lub z wykorzystaniem bezpośredniego połączenia pomiędzy urządzeniami posiadającymi łącza podczerwieni lub obsługującymi standard Bluetooth.

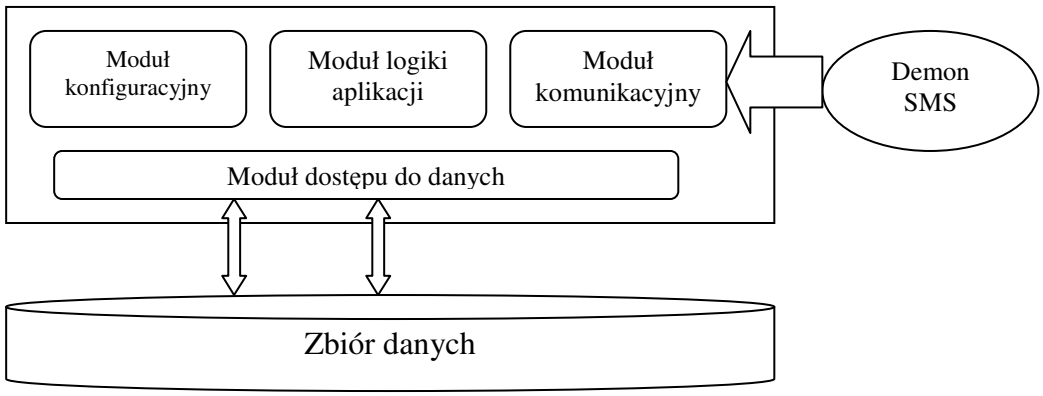
Ogólny schemat systemu wraz z metodami komunikacji ilustruje rys 3.1.



Rysunek 3.1 Ogólny schemat systemu

3.2 Aplikacja mobilna

Program stanowiący główną część systemu i odpowiadający za realizację podstawowych funkcji systemu. Aplikacja mobilna jest w dużym stopniu niezależna i spełnia większość funkcjonalności systemu nawet w przypadku braku komunikacji z pozostałymi elementami rozwiązania. Zainstalowana jest na urządzeniu mobilnym typu smartphone lub PDA-Phone i uruchamiana na żądanie użytkownika. Aplikacja mobilna składa się z trzech głównych części: Zbioru danych, Modułu operacyjnego i Demona SMS.



Rysunek 3.2: Schemat Aplikacji mobilnej

Zbiór danych zawiera informacje o przechowywanych listach zakupów i bazę produktów, które mogą wejść w skład listy. Informacje te są przechowywane w tabelach:

Products – baza danych produktów,

*ShoppingList*s – informacje o listach zakupów,

ShoppingListElements – lista produktów do zakupu.

Moduł operacyjny odpowiada za realizację logiki aplikacji. Komunikacja z użytkownikiem odbywa się za pomocą interfejsu graficznego wyświetlanego na ekranie telefonu. Moduł operacyjny możemy podzielić na podmoduły ze względu na zadania, które wykonują.

Moduł dostępu do danych – odpowiada za komunikację ze zbiorem danych i dba, o to by baza danych produktów była jak najmniejsza.

Moduł logiki aplikacji – odpowiada za zarządzanie listami zakupów i komunikację z użytkownikiem.

Moduł komunikacyjny – zapewnia połączenie z aplikacjami mobilnymi na innych urządzeniach, oraz serwerami uaktualnień i produktów.

Moduł konfiguracyjny – umożliwia konfigurację całej aplikacji przez interfejs w telefonie i za pomocą Aplikacji konfiguracyjnej na tradycyjnym komputerze PC, gdy urządzenie mobilne jest do niego podłączone.

3.3 Demon SMS

Demon SMS jest podprogramem filtrującym przychodzące na urządzenie mobilne wiadomości SMS. Zadaniem demona jest przechwycenie wiadomości wysłanych przez inne urządzenia należące do systemu, a następnie uruchomienie właściwej aplikacji i przekazanie otrzymanej wiadomości SMS do modułu komunikacyjnego. Aplikacja przetwarza otrzymaną wiadomość i informuje użytkownika o rezultatach. Dzięki wykorzystaniu demona właściwa aplikacja może pozostać nieaktywna, aż do uruchomienia przez użytkownika lub nadejścia komunikatu SMS.

3.4 Serwer Produktów

Zadaniem serwera produktów jest przyjmowanie zamówień od użytkowników. Aplikacja mobilna kontaktuje się z serwerem produktów i przekazuje mu listę zakupów, którą użytkownik chce zamówić. Serwer stara się dopasować elementy z zamówienia z asortymentem dostępnym w sklepie. Jeżeli więcej niż jeden produkt pasuje do elementu zamówienia serwer prosi użytkownika, za pośrednictwem aplikacji mobilnej, o wybranie konkretnego produktu z listy sugerowanych. Serwer może też samodzielnie wybrać produkty pasujące do listy otrzymanej listy zakupów zgodnie z zadanymi przez użytkownika kryteriami. Serwer produktów zapamiętuje profil użytkownika i wybrane przez niego produkty i korzysta z nich przy dobieraniu kolejnych zamówień. Żądania przychodzące od klientów powinny być obsługiwane w jak najkrótszym czasie, gdyż mogą wymagać aktywnej interakcji ze strony użytkownika aplikacji mobilnej.

W ramach serwera produktów można wyróżnić następujące elementy:

Baza danych – gdzie przechowywane są informacje o asortymencie sklepu, realizowanych zamówieniach i profilach klientów.

Moduł dostępu do danych – odpowiadającego za wykonywanie operacji bazodanowych niezależnie od działania procesu głównego i zapewnia synchronizacje dostępu do danych. Wstępnie przetworzone wyniki przekazywane są dalej.

Proces główny – zarządza aplikacjami mobilnymi aktualnie połączonymi z serwerem produktów. Utrzymuje informacje o połączeniach oraz stanie serwera i obsługuje nadchodzące połączenia uruchamiając dla nich oddzielne wątki.

Wątki pomocnicze – każda aplikacja mobilna łącząca się z serwerem produktów obsługiwana jest przez jeden wątek. Zadaniem wątku jest obsługa otrzymywanych żądań. Wątek kończy się po przyjęciu zlecenia.

3.5 Serwis Uaktualnień

Aplikacja mobilna łącząca się z serwisem uaktualnień ma możliwość pobrania nowszej wersji oprogramowania, która następnie może być automatycznie zainstalowana na urządzeniu mobilnym. Serwis udostępnia też aktualną listę produktów oferowanych przez dany sklep, która może być pobrana przez aplikację mobilna jako dodatek do posiadanej bazy produktów. Korzystanie ze ściągniętej z serwisu listy produktów ułatwia późniejsze dokonywanie zamówień, gdyż lista zakupów stworzona za pomocą produktów z tej listy bezpośrednio przenosi się na konkretne produkty w asortymencie sklepu. Funkcjonalność serwisu uaktualnień może być oferowana na platformie serwera produktów.

3.6 Aplikacja konfiguracyjna

Aplikacja konfiguracyjna instalowana jest na tradycyjnym komputerze klasy PC, do którego następnie można podłączyć urządzenie mobilne, na którym zainstalowana jest Aplikacja mobilna. Aplikacja konfiguracyjna jest programem narzędziowym za jej pomocą można dokonywać zmian

w ustawieniach programu mobilnego. Wykorzystanie aplikacji konfiguracyjnej ma ułatwić użytkownikowi wprowadzenie danych do urządzenia dzięki wykorzystaniu pełnowymiarowej klawiatury i myszy.

W skład aplikacji konfiguracyjnej wchodzą dwa moduły odpowiadające za realizację głównych funkcji programu.

Moduł komunikacyjny – zapewnia połączenie z urządzeniem mobilnym, które zostało podłączone do komputera stacjonarnego za pomocą łącza podczerwieni, Bluetooth lub kabla USB. Moduł komunikacyjny nawiązuje połączenie z urządzeniem i pobiera z niego zawartość przechowywanego zbioru danych i opcji konfiguracyjnych.

Moduł edycji – umożliwia wygodną edycję pobranego zbioru danych i ustawienie opcji konfiguracyjnych. Zmienione dane są następnie ładowane z powrotem do pamięci urządzenia mobilnego za pomocą modułu komunikacyjnego.

4. Implementacja

4.1 Aplikacja mobilna

4.1.1 Wykorzystane technologie

Podczas implementacji aplikacji mobilnej zostały wykorzystane następujące technologie.

- Aplikacja przeznaczona jest do wykorzystania na telefonach komórkowych typu smartphone z zainstalowanym mobilnym systemem operacyjnym Windows Mobile 5.0 lub nowszym. Technologia Windows Mobile została wybrana ze względu na jej rosnącą popularność, bogatą gamę narzędzi programistycznych i wsparcie dla integracji tworzonych aplikacji ze stacjonarnymi systemami Windows.

- Program został napisany przy użyciu języka C#. Zdecydowana większość kodu aplikacji wykorzystuje kod zarządzany i jest uruchamiana za pośrednictwem .NET Compact Framework w wersji 2.0. Wykorzystanie zarządzanego kodu i CLR gwarantuje, że aplikacja będzie działać i wyglądać tak samo na różnych urządzeniach. Compact Framework umożliwia też korzystanie z mechanizmów zdarzeń, wyjątków, kontroli typów i zarządzania pamięcią (odśmiecanie). Realizacja niektórych funkcjonalności wymagała użycia wywołań systemowych P/Invoke, które działają w trybie natywnym.

- Dane aplikacji przechowywane są w plikach XML. W trakcie prac nad systemem zrezygnowano z zastosowania bazy danych dla urządzeń mobilnych (MS SQL Server Mobile Edition), ponieważ zakres informacji przechowywanych przez aplikację jest stosunkowo niewielki. Nie istnieją skomplikowane powiązania pomiędzy tabelami ani nie ma potrzeby synchronizowania danych z zewnętrzną tradycyjną bazą danych. Wykorzystanie XML powoduję dłuższy czas dostępu do danych, ale cała aplikacja zajmuje mniej pamięci operacyjnej. Nie bez znaczenia jest też to, że w przypadku wykorzystania komercyjnego aplikacji nie potrzebna jest licencja na bazę danych.

- Interfejs użytkownika zbudowany został ze standardowych kontrolek Windows.Forms dostępnych w bibliotekach .NET Compact Framework, które w razie potrzeby zostały rozszerzone o dodatkową funkcjonalność. Wykorzystanie standardowych kontrolek jest zalecane w środowisku mobilnym ze względu na ich dodatkową optymalizację dla urządzeń przenośnych dysponujących ograniczonymi zasobami. Dodatkowo standardowe kontrolki ułatwiają użytkownikowi korzystanie z aplikacji sprawiając wrażenie, że aplikacja jest częścią systemu operacyjnego.

- Jako środowisko programistyczne wykorzystano Visual Studio .NET 2005 z zainstalowanym rozszerzeniem Windows Mobile 5.0 SDK for smartphone. Platforma została wykorzystana zgodnie z wydziałową licencją MSDN Academic Alliance.

- W trakcie implementacji systemu wykorzystano darmowy program do emulacji urządzeń mobilnych Device Emulator. Na początku prac wykorzystywana była wersja 1.0, a następnie nowo wydana wersja 2.0, w której m.in. znacznie poprawiono wydajność emulatora. Wybrany emulator umożliwia nie tylko symulację wyglądu i interfejsu

urządzenia mobilnego, ale stanowi swoistą wirtualną maszynę. Emulacji podlega również procesor (ARM), pamięć flash i częściowo usługi telefoniczne.

4.1.2 Podział na biblioteki

Wszystkie klasy i kontrolki wykorzystywane w aplikacji przechowywane są w jednym z trzech plików:

- BasicLib.dll – Biblioteka zawiera klasy i funkcje wykorzystujące wywołania systemowe P/Invoke, realizujące funkcjonalność nie oferowaną przez .NET Compact Framework. Funkcje te są następnie wywoływane przez pozostałe moduły napisane już w kodzie zarządzanym.

- ShoppingListControls.dll – W skład biblioteki wchodzą elementy odpowiedzialne za interfejs użytkownika, oraz niektóre fragmenty logiki aplikacji. Znajdują się w niej głównie formatki i kontrolki. Klasy podzielone są na dwie przestrzenie nazw – `ShoppingList.Forms` i `ShoppingList.ShoppingListControls`.

- ShoppinList.exe – Główny plik programu ładowany przy starcie aplikacji. Zawiera moduł dostępu do źródeł danych, klasy realizujące logikę aplikacji i komunikację z urządzeniami i usługami zewnętrznymi.

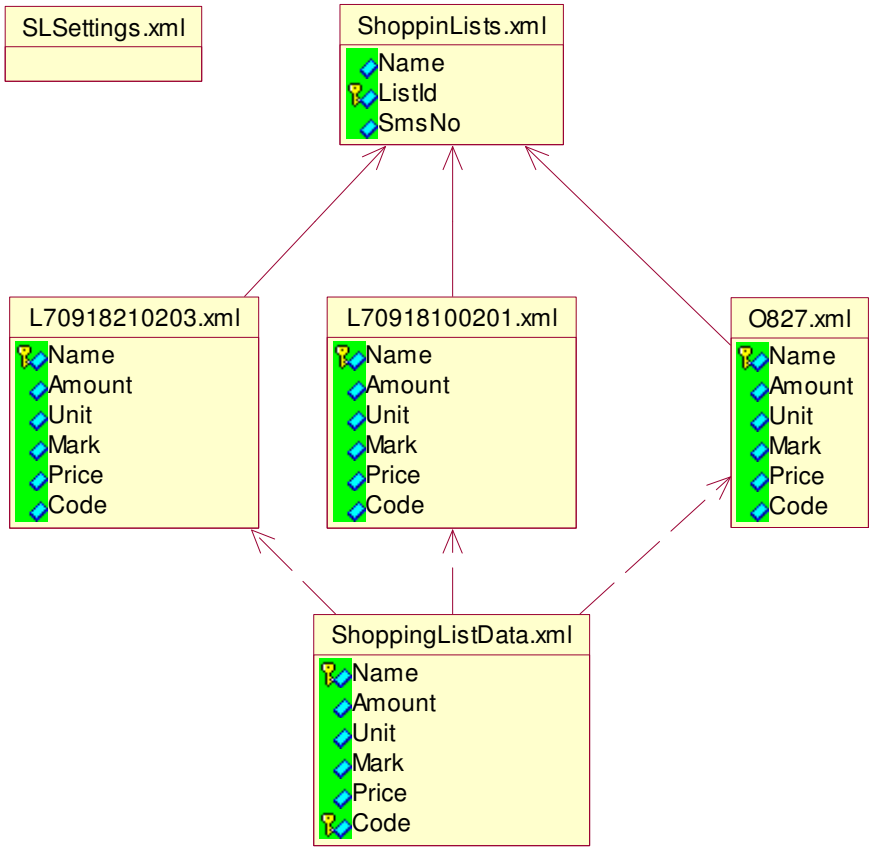
- Katalog Fonts – zawiera ikony w postaci plików Portable Network Graphics (PNG) wykorzystywane w menu aplikacji.

4.1.3 Zbiór danych

Wszystkie dane aplikacji przechowywane są w plikach XML w katalogu \Application Data\ShoppingList\. W dodatku B znajdują się pliki XML Schema Definition (XSD), na podstawie których tworzone są poszczególne pliki. Schemat źródła danych wraz z zależnościami przedstawiony jest na rysunku 4.1.

Pliki XML i ich znaczenie:

- `ShoppingLists.xml` – przechowuje informacje o dostępnych w aplikacji listach zakupów
    - Name – nazwa listy nie dłuższa niż 15 znaków.
    - ListId – identyfikator listy.
    - SmsNo – liczba SMS-ów otrzymanych dla danej listy.
- `LYMMDDHHMMSS.xml` – Istnieje wiele plików o tym schemacie nazwy. Każdy plik odpowiada jednemu rekordowi w pliku `ShoppingLists.xml` o polu ListId zgodnym z nazwą pliku XML. Pliki te przechowują produkty znajdujące się na lokalnej liście zakupów. Nazwa pliku powstaje przez połączenie litery „L”, daty i czasu utworzenia listy.
    - Name – nazwa produktu nie dłuższa niż 30 znaków.
    - Amount – ilość produktu, jaką należy kupić.
    - Unit – jednostka wagi lub pojemności.
    - Mark – oznaczenie, czy produkt został już kupiony.



Rysunek 4.1: Schemat zbioru danych i plików XML

- Price – cena jednostkowa produktu.
- Code – kod produktu.

- `ONNN.xml` – Może istnieć wiele plików o tym schemacie nazwy. Każdy plik odpowiada jednemu rekordowi w pliku `ShoppingLists.xml` o polu zgodnym z nazwą pliku XML. Pliki te przechowują tymczasowe listy produktów otrzymane ze źródeł zewnętrznych.

- `ShoppingListData.xml` – przechowuje specjalną listę produktów pobraną z serwera produktów.

- `SLSettings.xml` – przechowuje informacje o aktualnej konfiguracji aplikacji.

Za odczytywanie i zapisywanie informacji do plików XML odpowiedzialna jest klasa `DataStorage`. W systemie istnieje tylko jeden obiekt tej klasy tworzony przy uruchomieniu aplikacji i dostępny dla innych klas jako statyczny atrybut głównej klasy `Program`. Kontrolki i klasy logiki aplikacji odwołują się do obiektu klasy `DataStorage` za każdym razem, gdy chcą pobrać, zapisać, zmodyfikować czy usunąć jakiekolwiek informacje. Aby usprawnić pracę z danymi i ograniczyć liczbę kosztownych zapisów i odczytów klasa `DataStorage` przechowuję dane z ostatnio otwartych plików w pamięci, realizując tym samym prosty mechanizm buforowania. Gdy klasy logiki aplikacji proszą o dane z plików XML, klasa `DataStorage` sprawdza najpierw czy dany obiekt nie został wcześniej wczytany. Jeśli był, to przekazuje obiekt z pamięci, jeśli nie to odczytuje dane z pliku XML. W ustawieniach aplikacji można określić ile ostatnio wczytanych obiektów może być przechowywanych w pamięci przez klasę

DataStorage. Domyślną wartością jest 0, czyli liczba ograniczona tylko rozmiarem dostępnej pamięci operacyjnej.

4.2 Demon SMS

Realizacja funkcjonalności Demona SMS na platformie Windows Mobile 5.0 jest bardzo prosta. W tym celu można wykorzystać klasę MessageInterceptor znajdującą się w przestrzeni nazw Microsoft.WindowsMobile.PocketOutlook.MessageInterception będącej częścią standardowej biblioteki .NET Compact Framework. Klasa ta pozwala zarejestrować w systemie operacyjnym zdarzenie, które w momencie przyjścia SMS-a spełniającego podane kryteria uruchomi wybrany program przekazując do niego treść otrzymanej wiadomości. Dzięki temu mechanizmowi program może pozostać całkowicie nieaktywny aż do momentu przyjścia odpowiedniej wiadomości.

```
MessageInterceptor rule;
ShoppingList mainwindow = new ShoppingList();

//Rejestracja zdarzenia
//Sprawdzamy czy zdarzenie nie jest już zarejestrowane
if (args.Length == 0 &&
!MessageInterceptor.IsApplicationLauncherEnabled("ShoppingList"))
{
     rule = new MessageInterceptor(InterceptionAction.NotifyAndDelete);
     MessageCondition condition = new
     MessageCondition(MessageProperty.Subject,
                    MessagePropertyComparisonType.StartsWith,
                    Settings.SmsOpener,
                    false);
     rule.MessageCondition = condition;
     rule.EnableApplicationLauncher("ShoppingList",
     Settings.ProgramPath, "EventFired");

     rule.MessageReceived += new
     MessageInterceptorEventHandler(mainwindow.msgReceived);
}
else
{
     //Podczepienie handler'ow
    MessageInterceptor messageInterceptor = new
    MessageInterceptor("ShoppingList");
    messageInterceptor.MessageReceived += new
    MessageInterceptorEventHandler(mainwindow.msgReceived);
}
```

Zdarzenie rejestrowane jest tylko przy pierwszym uruchomieniu aplikacji. Przy kolejnych uruchomieniach podczepiamy tylko odpowiednie funkcje obsługujące, na wypadek gdyby podczas działania aplikacji pojawił się SMS, który powinniśmy przechwycić. Parametr InterceptionAction.NotifyAndDelete oznacza, że po przekazaniu SMS-a do aplikacji zostanie on wykasowany ze skrzynki z wiadomościami w telefonie. Na zmiennej condition typu MessageCondition przechowywany jest warunek, jaki musi spełnić SMS, aby mógł zostać przechwycony. W tym przypadku treść SMS-a musi zaczynać się od wartości przechowywanej na zmiennej Settings.SmsOpener. Domyślnie aplikacja przechwytuje wszystkie wiadomości zaczynające się ciągiem znaków „SL:", ale można tę wartość zmienić w konfiguracji. Przechwycony SMS przekazywany jest do funkcji mainwindow.msgReceived, która uruchamia dla otrzymanego SMS-a funkcję ParseSMS z klasy SMSSender. Jeśli uda się poprawnie sparsować wiadomość wykonywana jest odpowiednia dla danego SMS-a akcja. Jeśli

wystąpi błąd w parsowaniu, to SMS jest przekazywany do systemu operacyjnego na wypadek gdyby przechwycona wiadomość tylko przypadkowo zaczynała się od ciągu znaków „SL:”.

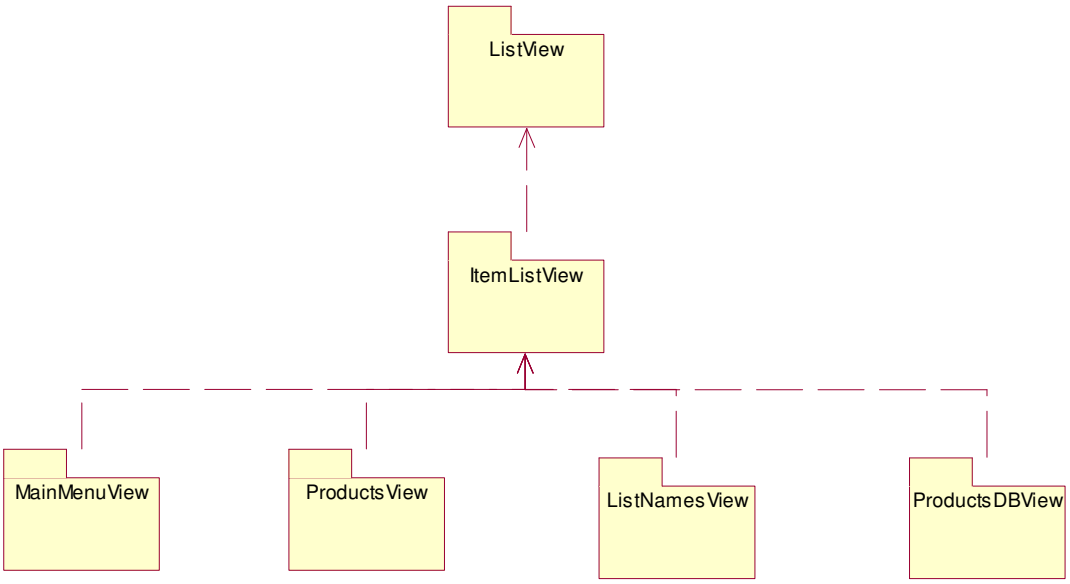
4.3 Interfejs Użytkownika

Interfejs użytkownika stworzono wykorzystując kontrolki Window Forms dla urządzeń mobilnych będące częścią standardowej biblioteki klas dołączonych do .NET Compact Framework.

4.3.1 Kontrolki

Najważniejszym elementem interfejsu użytkownika są kontrolki odpowiedzialne za wyświetlanie danych przechowywanych w plikach XML. W pierwszej wersji implementacji interfejsu wykorzystano do tego standardowe kontrolki `DataGrid` w wersji dla urządzeń mobilnych. Główne zalety tych kontrolek to:

- Możliwość podłączenia kontrolki do źródła danych (Data Binding). Kontrolka automatycznie zmienia prezentowane dane, gdy zmieni się zawartość zbioru danych.
- Umożliwia edycję, dodawanie, usuwanie danych z podłączonego źródła danych.
- Wybieranie elementów i przekazywanie ich wartości do innych kontrolek.
- Sortowanie danych.
- Automatyczne dostosowanie prezentacji informacji do struktury otrzymanych danych.
- Współpraca z plikami XML i mobilną bazą danych MS SQL Server Mobile Edition.



Rysunek 4.2: Schemat dziedziczenia kontrolek interfejsu użytkownika

Interfejs zbudowany z użyciem kontrolek `DataGrid` okazał się jednak mało wydajny, z uwagi na duży rozmiar tych obiektów powodujący znaczący narzut na czas ich inicjacji. Ze względu na swoją uniwersalność kontrolki `DataGrid` sprawdzają wiele warunków i założeń, aby możliwie jak najlepiej dostosować się do źródła danych. Przy ładowaniu danych z pliku XML do kontrolki,

każdy wiersz jest walidowany pod względem poprawności. Wszystkie te elementy powodują, że ładowanie kontrolki trwa stosunkowo długo (nawet do 2-3 sek przy niewielkich zmianach zbioru danych).

W celu poprawienia wydajności interfejsu użytkownika w miejsce kontrolek `DataGrid` wykorzystano kontrolkę `ListView`. Kontrolki te przygotowane są z myślą o prezentacji statycznych danych w postaci prostej listy, małych i dużych ikon albo w trybie szczegółowym. Wypełnioną danymi kontrolkę można dowolnie przełączać pomiędzy trybami wyświetlania. Kontrolki `ListView` są znacznie mniejsze niż kontrolki `DataGrid`, ale też oferują znacznie mniejszy zakres funkcjonalności. W aplikacji wykorzystywana jest kontrolka `ItemListView`[1] dziedzicząca po kontrolce `ListView` i rozszerzająca jej możliwości o uproszczoną funkcjonalność podłączania do źródeł danych (DataBinding) specjalnie przygotowaną do współpracy z klasą `DataStorage`.

Po klasie `DataListView` dziedziczą pozostałe kontrolki wykorzystywane w aplikacji do prezentacji różnych list. Diagram dziedziczenia przedstawia Rysunek 4.2, a na rysunku 4.3 zaprezentowane są dwa tryby wyświetlania danych w kontrolce.

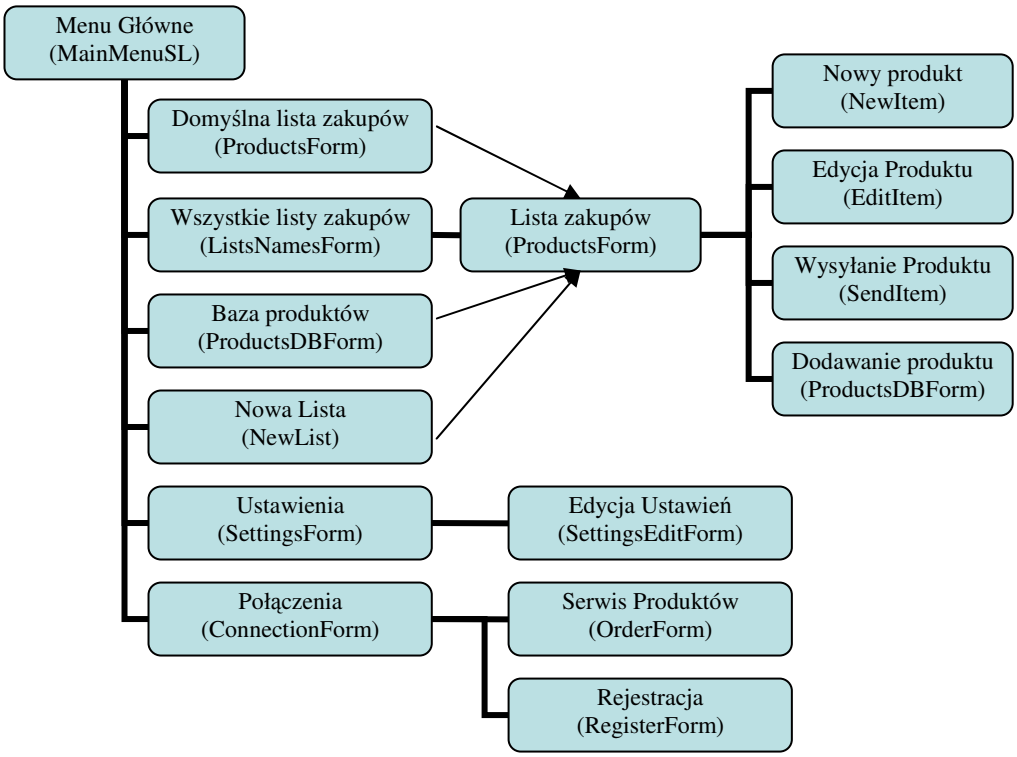
Rysunek 4.3 Główne menu aplikacji i przykładowa lista zakupów

4.3.2 Formatki

Wszystkie wykorzystane w aplikacji formatki dziedziczą po klasie Windows.Forms. W każdej formatce można wyróżnić dwie części: obszaru wyświetlania i menu. Obszar wyświetlania realizowany jest przez kontrolki dziedziczące po klasie ItemListView i odpowiada za prezentację danych lub ich edycję. Menu jest realizowane przez kontrolki standardowej klasy MainMenu. Menu jest podzielone na dwie części lewą i prawą, a uruchomienie menu odbywa się przez wciśnięcie odpowiednio lewego lub prawego klawisza szybkiego dostępu. Na każdej formatce lewa

[1] Kontrolka powstała na podstawie przykładu kontrolki BindableListView [9].

część menu odpowiada za uruchomienie najczęściej stosowanej akcji dla danej formatki. Prawa część umożliwia rozwinięcie listy akcji dodatkowych. Na rysunku 4.4 przedstawiono ogólny schemat interfejsu użytkownika i powiązań między formatkami.



Rysunek 4.4: Schemat interfejsu użytkownika

Poszczególne elementy głównego menu odpowiadają za:

- Domyślna lista zakupów – pozwala na bezpośredni dostęp do aktualnie wykorzystywanej listy zakupów. Domyślną listę można określić w ustawieniach aplikacji.
- Wszystkie listy zakupów – Wyświetla listę nazw wszystkich istniejących w systemie list. Umożliwia dodawanie, usuwanie, zmianę nazwy i wysyłanie listy do innych użytkowników systemu.
- Baza produktów – zawiera listę ściągniętych z serwisu uaktualnień produktów dostępnych w sklepie. Elementy z tej listy można następnie dodawać do własnych spersonalizowanych list zakupów. Można również dodawać do niej własne produkty do późniejszego wykorzystania.
- Nowa Lista – umożliwia szybkie stworzenie nowej listy.
- Ustawienia – wyświetla listę parametrów konfiguracyjnych aplikacji.
- Połączenia – umożliwia połączenie się z serwerem produktów, serwisem uaktualnień i zarządzanie funkcjami przesyłania list.

4.4 Serwer produktów i serwis uaktualnień

Ponieważ głównym tematem pracy są aplikacje mobilne zaprezentowana implementacja serwera produktów stanowi tylko prototyp prawdziwego systemu do realizacji zamówień. Przy pracy nad prototypem skupiłem się na implementacji komunikacji pomiędzy smartphonem a serwerem.

4.4.1 Wykorzystane technologie

Przy implementacji serwera produktów i serwisu uaktualnień wykorzystano następujące technologie (oprócz wymienionych w rozdziale 5.1.1)

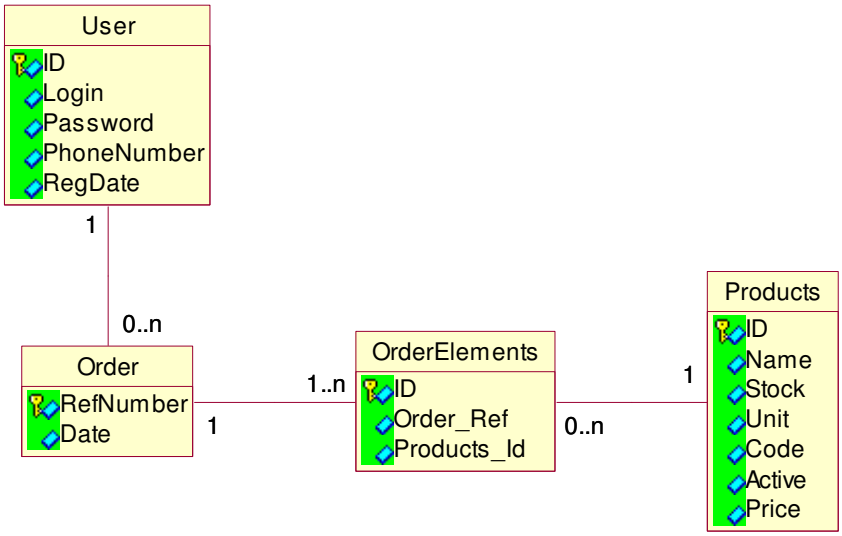
- Dane zapisywane są w bazie danych Microsoft SQL Serwer 2005 Express Edition. Jest to darmowa wersja bazy przeznaczona dla niewielkich aplikacji oraz jako zastępstwo dla pełnej wersji bazy danych podczas tworzenia aplikacji. Programy napisane z wykorzystaniem MS SQL Express Edition można instalować na pełnej wersji bazy danych Microsoftu.

- Aplikacja działa na serwerze WWW Internet Information Services (IIS) 6.0.

- Do komunikacji z aplikacją mobilną wykorzystano technologię usług sieciowych. Technologia ta dzięki zastosowaniu plików XML i protokołu SOAP umożliwia zdalne wywoływanie usług przez zewnętrzne systemy i aplikacje niezależnie od platformy na jakiej działają.

4.4.2 Baza danych

Dane serwera produktów i serwisu uaktualnień przechowywane są w prostej bazie danych przedstawionej na rysunku 4.5.



Rysunek 4.5: Schemat bazy danych serwera produktów

Poszczególne tablice odpowiadają za:

- User – przechowuje podstawowe informacje o użytkowniku, który może korzystać z serwisu.

- Products – zawiera dane o produktach dostępnych w sklepie.

- Order i Order Elements – odpowiadają za przechowywanie zleceń użytkowników.

4.4.3 Usługi sieciowe

Cała komunikacja pomiędzy serwerem a aplikacją mobilną odbywa się za pośrednictwem usług sieciowych. Technologia ta została wybrana ze względu na jej uniwersalność. Aplikacja mobilna nie musi być stale połączona z serwerem, aby poprawnie działać. Dodatkowo długotrwałe utrzymywanie połączenia z siecią Internet mogłoby wiązać się z dodatkowymi kosztami dla użytkownika telefonu. Do realizacji założonej funkcjonalności wystarczą krótkie sesje połączeniowe realizowane przy korzystaniu z usług sieciowych. Wykorzystanie tej technologii umożliwiło również zintegrowanie funkcjonalności zamawiania produktów z aplikacją przechowującą te listy zakupów. Nie ma potrzeby uruchamiania oddzielnych programów do łączenia się ze sklepem (np. przeglądarki internetowej), a cały proces komunikacji przebiega w tle.

Funkcjonalność serwera produktów i serwisu uaktualnień realizowany jest przez trzy usługi sieciowe:

- `RegisterUser` – służy do rejestrowania nowego użytkownika w systemie realizującym zamówienia.

- `PlaceOrder` – realizuje funkcjonalność zamawiania produktów z przechowywanych w aplikacji mobilnej list zakupów.

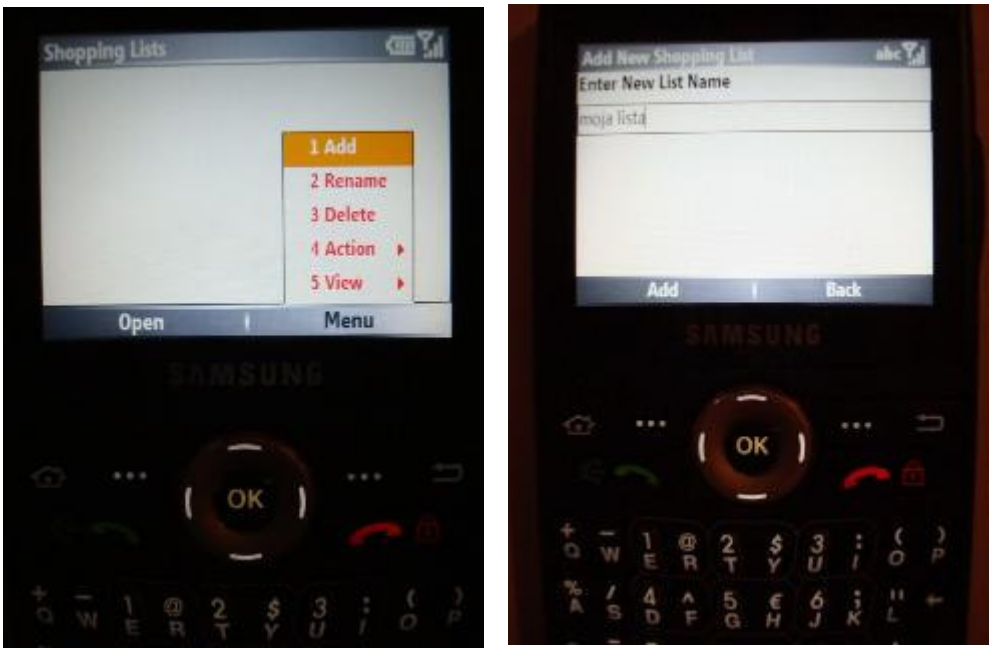
- `GetOfferedProducts` – umożliwia pobranie z serwisu listy dostępnych produktów, z których następnie można tworzyć własne listy zakupów.

5. Przypadki użycia

5.1 Listy zakupów

5.1.1 Tworzenie nowej listy

Pracę z aplikacją zaczyna się zwykle od stworzenia własnej listy zakupów. Aby dodać nową listę zakupów należy z menu głównego wybrać ikonę „New List", albo będąc w formatce o nazwie *Shopping Lists* z menu wybrać opcję „Add". Pojawi się okno dialogowe z prośbą o podanie nazwy tworzonej listy. Po wpisaniu nazwy należy zatwierdzić operację klawiszem „Add". Można też zrezygnować ze stworzenia listy wciskając klawisz „Back". Rysunek 5.1 ilustruje przebieg kolejnych operacji. Nowoutworzona lista widoczna będzie w spisie dostępnych list (formatka *Shopping Lists*). Jeśli nową listę utworzono za pomocą ikony „New List" to automatycznie otworzy się pusty podgląd artykułów listy, umożliwiając rozpoczęcie dodawania produktów.



Rysunek 5.1: Procedura zakładania listy zakupów

5.1.2 Zmiana nazwy listy

Zmiany nazwy listy dokonuje się przez wskazanie za pomocą klawiszy nawigacyjnych listy w formatce *Shopping Lists* i wybranie z menu opcji „Rename". Pojawi się okno dialogowe, w którym można edytować nazwę wybranej listy. Zmianę należy zatwierdzić klawiszem „Rename". Wciśnięcie klawisza „Back" spowoduje odrzucenie wszelkich zmian. Funkcjonalność edycji nazwy listy dostępna jest też z formatki zawierającej podgląd artykułów listy po wybraniu opcji „Menu-> List->Rename".

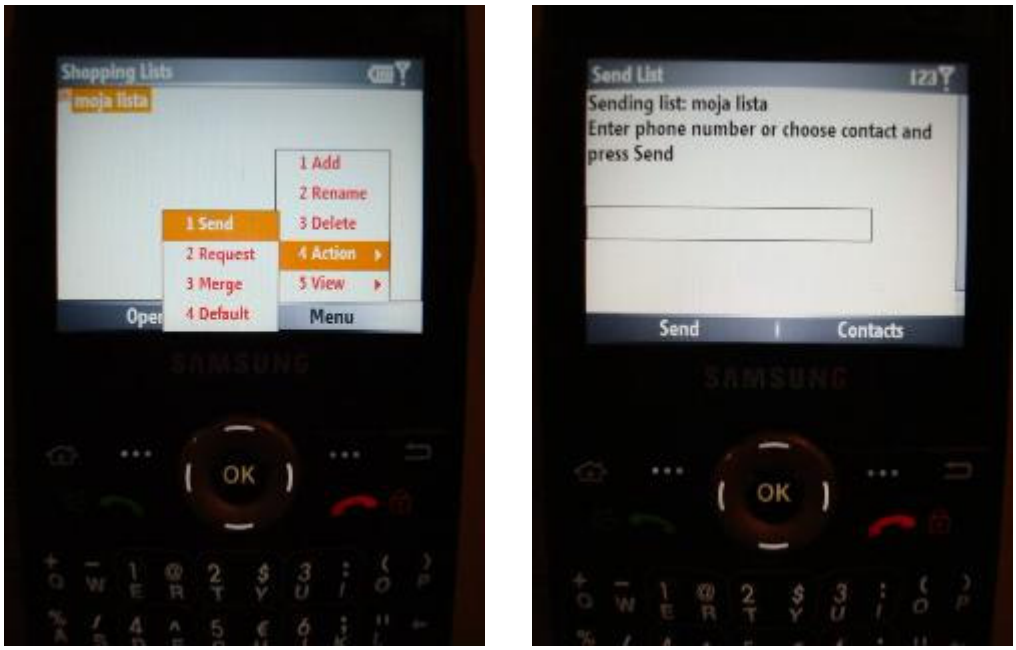
5.1.3 Usuwanie listy

Usuwanie listy produktów odbywa się przez podświetlenie listy do usunięcia w formatce *Shopping Lists* i wybranie z menu opcji Delete. Pojawi się okno dialogowe z prośbą o potwierdzenie operacji. Usunięcie listy należy zatwierdzić wciskając przycisk „Yes” lub zrezygnować wciskając przycisk „No”. Usuwanie listy jest też możliwe z formatki prezentującej podgląd artykułów listy przez wybranie opcji „Menu->List->Delete”.

5.1.4 Wysyłanie listy za pośrednictwem SMS

Jedną z najciekawszych funkcjonalności systemu jest możliwość wysyłania stworzonych list zakupów do innych użytkowników systemu korzystających z własnych urządzeń mobilnych, przy założeniu, że korzystają z tej samej wersji aplikacji mobilnej.

Aby wysłać komuś naszą listę zakupów, należy z głównego menu otworzyć spis wszystkich dostępnych list (Shopping Lists). Następnie zaznaczyć listę, którą chcemy wysłać i z menu dostępnego pod prawym klawiszem funkcyjnym wybrać opcję „Action->Send”. Po wybraniu akcji otworzy się okno dialogowe z prośbą o podanie numeru telefonu, na jaki należy wysłać listę. Można podać numer ręcznie lub wybrać osobę z listy kontaktów telefonu komórkowego (wciskając prawy klawisz szybkiego dostępu). Po wybraniu numeru zatwierdzamy całą operację lewym klawiszem szybkiego dostępu („Send”). Całą operację ilustruję rysunek 5.2. Wysyłanie listy jest też możliwe z formatki prezentującej artykuły na liście przez wybranie opcji „Menu->List->Send”.



Rysunek 5.2: Procedura wysyłania listy zakupów

Wciśnięcie klawisza „Send” powoduje wywołanie metody `SendShoppingList` z klasy `SenderSMS`, której zadaniem jest wysłanie wybranej listy. Wszystkie wysyłane SMS mają analogiczną budowę zgodną z wzorcem:

```
SL:<Akcja>%<identyfikator>%<nr SMS-a>%<liczba SMS-ow>%<opcja>%<body>%
```

Poszczególne elementy oznaczają:

- <Akcja> – określa akcję realizowaną przez SMS. W tym przypadku jest to wysyłanie listy oznaczone przez znak „L”.

- <identyfikator> – trzycyfrowy losowy identyfikator wysyłanej listy. Generowany jest za każdym razem przy wysyłaniu listy i ma zapobiec wymieszaniu się list w przypadku, gdy w tym samym momencie przychodzą do nas dwie różne listy zakupów z dwóch różnych źródeł.

- <nr SMS-a> – numer wysyłanego SMS-a w ramach listy.

- <liczba SMS-ów> – całkowita liczba SMS-ów wysłana w ramach listy.

- <opcja> – pole dodatkowe. Przy wysyłaniu list wypełnione maksymalnie piętnastoma pierwszymi znakami z nazwy listy.

- <body> – treść zawierająca rekordy odpowiadające produktom z wysyłanej listy.

Przy wysyłaniu listy dla każdego produktu wysyłane są tylko pola Code (identyfikator), Name (nazwa), Amount (ilość) i Unit (jednostka). W jednym SMS-ie jest wysyłana maksymalna liczba produktów, jaką uda się zmieścić w treści wiadomości.

Aplikacja, do której przychodzą komunikaty zawierające listę zakupów, parsuje otrzymane dane i tworzy listę o ListID równym „O" + <identyfikator> i nazwie otrzymanej z pola <opcje>. Produkty z kolejnych SMS-ów zapisywane są do stworzonej tak listy tymczasowej. Po otrzymaniu wszystkich wiadomości generowany jest lokalny identyfikator listy, a użytkownik telefonu informowany jest o tym specjalnym komunikatem.

5.1.5 Wysyłanie zapytania o listę zakupów do innych użytkowników

Aplikacja umożliwia również wysyłanie komunikatu do innych użytkowników systemu z prośbą o przysłanie własnych list zakupów.

Dostęp do tej funkcjonalności uzyskuje się przez wybranie w formatce *Shopping Lists* opcji „Menu->Actions->Request". Po wybraniu akcji otworzy się okno dialogowe z prośbą o podanie numeru telefonu, na jaki należy wysłać komunikat. Można podać numer ręcznie lub po wciśnięciu klawisza „Contacts" wybrać osobę z listy kontaktów telefonu komórkowego. Po wybraniu numeru zatwierdzamy całą operację klawiszem „Send". Aby zrezygnować z wysyłania trzeba wcisnąć klawisz końca rozmowy.

Użytkownik, do którego wysłaliśmy wiadomość, informowany jest o nim stosownym komunikatem. Jeśli miał zdefiniowaną listę domyślną może ją natychmiast wysłać lub wybrać inną listę do wysłania. W ustawieniach aplikacji można ustawić opcję szybkiej odpowiedzi („Request Replays" ustawione na „Auto"), która automatycznie wysyła domyślną listę zakupów.

5.1.6 Scalanie dwóch list

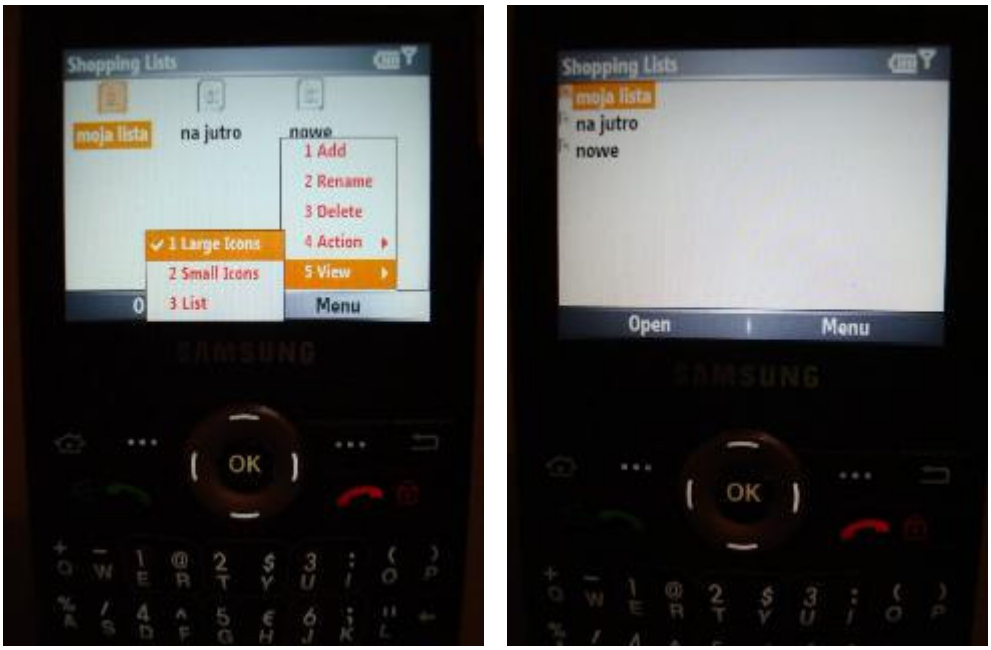
Łączenie dwóch list odbywa się przez zaznaczenie w formatce *Shopping Lists* pierwszej listy zakupów i użycie opcji „Menu->Actions->Merge". Po wybraniu akcji pojawi się nowe okno z listami zakupów, z którego należy wybrać drugą listę do połączenia. Powstała lista zachowa nazwę pierwszej wybranej listy.

W przypadku, gdy na obu listach występują te same pozycje zachowywane są odpowiednie wartości z pierwszej wybranej listy z wyjątkiem pola „Amount" (Ilość), które zwiększane jest o wartość tego pola z listy drugiej.

Dostęp do tej funkcjonalności możliwy jest też bezpośrednio z formatki z podglądam produktów przez wybranie opcji „Menu->List->Merge". Otwarta lista traktowana jest jako pierwsza lista do połączenia.

5.1.7 Zmiana wyświetlania list zakupów

Formatka *Shopping Lists* udostępnia trzy tryby wyświetlania zawartości: duże ikony, małe ikony i lista. Zmiana trybu wyświetlania odbywa się przez przejscie do opcji „Menu->View" i wybranie żądanego trybu („Large Icons", „Small Icons", „List"). Przykłady dwóch trybów wyświetlania prezentuje rysunek 5.3.



Rysunek 5.3: Tryb wyświetlania dużych i małych ikon

5.1.8 Ustawienie listy jako domyślnej

Lista ustawiona jako domyślna może być otwierana bezpośrednio z menu głównego aplikacji przez wybranie ikony „Current List". Dodatkowo w przypadku otrzymania komunikatu z prośbą o wysłanie listy zakupów proponowane jest automatyczne wysyłanie listy domyślnej.

Aby wybrać listę domyślną należy w formatce *Shopping Lists* podświetlić żądaną listę i wybrać opcję „Menu->Actions->Default". Alternatywną metodą jest skorzystanie z opcji „Menu->Lists-> Default" z formatki z podglądem artykułów należących do listy.

5.2 Praca z listą zakupów

5.2.1 Dodawanie nowego produktu do listy

Dodawanie nowych produktów do listy zakupów możliwe jest z poziomu formatki z podglądem zawartości danej listy, do której przechodzimy podświetlając wybraną listę zakupów w formatce *Shopping Lists* i wciskając klawisz „Open" lub „Enter".

Do dodawania nowych elementów służy opcja „New" z menu dostępnego pod prawym klawiszem funkcyjnym. Po wybraniu akcji pojawi się nowa formatka (*New Item*), w której można podać nazwę artykułu, ilość jaką chcemy zakupić, jednostkę miary („g" – gramy, „kg" – kilogramy, „ml" – litry, „l" – litry, lub pustą), oraz cenę. Do poprawnego dodania produktu wymagane jest podanie wyłącznie nazwy. Operację zatwierdzamy klawiszem „Add". Po dodaniu elementu możliwe jest dodawanie kolejnych. Aby opuścić formatkę dodawania nowych produktów należy wcisnąć klawisz „Exit".

W przypadku, gdy produkt o danej nazwie istnieje już na liście to nie są dokonywane żadne zmiany.

5.2.2 Dodawanie produktów z bazy artykułów do listy

Aplikacja umożliwia też szybkie tworzenie list zakupów na podstawie wypełnionej wcześniej bazy artykułów.

Dodawanie produktów na podstawie bazy odbywa się poprzez wybranie opcji „Add" z menu. Po wybraniu akcji wyświetlana jest lista wszystkich artykułów dostępnych w bazie (formatka *Products Base*), z której możemy wybrać interesujące nas produkty. Przy pomocy klawiszy nawigacyjnych podświetlamy wybrany produkt i wciskamy klawisz „Add", aby dodać artykuł do listy. Po dodaniu elementu zostaje on zaznaczony na zielono i można dodawać kolejne artykuły. Tryb dodawania produktów można opuścić wciskając klawisz funkcyjny „Wstecz". Dodawanie artykułów z bazy ilustruje rysunek 5.4.

Rysunek 5.4: Dodawanie produktów na podstawie bazy artykułów

5.2.3 Edytowanie produktu

Użytkownik ma możliwość edycji wcześniej dodanych produktów. W tym celu należy w formatce prezentującej elementy listy zakupów zaznaczyć produkt do edycji i wybrać opcję „Edit" z menu. Pojawi się nowe okienko podobne do formatki *New Item,* w którym można zmienić nazwę, ilość, jednostkę i cenę produktu. Po wprowadzeniu zmian zatwierdzamy operację klawiszem „Update" lub rezygnujemy ze zmian przez wciśnięcie klawisza „Back". W przypadku, gdy zmienimy nazwę produktu i istnieje już artykuł o takiej nazwie to nie następują żadne zmiany, a użytkownik jest informowany o tym fakcie stosownym komunikatem.

5.2.4 Usuwanie produktu

Usuwanie elementu z listy odbywa się przez podświetlenie elementu do usunięcia w otwartej liście zakupów i wybranie z menu opcji „Delete". Pojawi się okno dialogowe, w którym można potwierdzić operację wciskając klawisz „Yes" lub zrezygnować z usunięcia produktu klawiszem „No".

5.2.5 Zmiana prezentacji listy zakupów

Użytkownik ma możliwość korzystania z jednego z dwóch trybów prezentowania produktów na liście. W trybie listy (Menu->View->List) prezentowane są tylko nazwy artykułów na liście. Pełne informacje o produktach dostępne są po wybraniu trybu „Szczegółowego" (Menu->View->Details). Domyślnie wykorzystywany jest tryb szczegółowy.

5.2.6 Zapisywanie produktu w bazie artykułów

Produkty, które zostały dodane do listy artykułów ręcznie nie są automatycznie zapisywane w bazie artykułów. Funkcjonalność dodawania artykułów do bazy może także okazać się przydatna do zapisywania artykułów z listy, którą otrzymano od innego użytkownika.

W celu zapisania wybranego artykułu do bazy należy po wskazaniu artykułu wybrać z menu opcję „Save". W przypadku, gdy dany produkt już istnieje na liście nie są wykonywane żadne zmiany.

5.2.7 Zaznaczanie zakupionych produktów

W celu ułatwienia zarządzania listą zakupów użytkownik ma możliwość zaznaczania artykułów, które już zostały kupione. Aby oznaczyć produkt jako zakupiony należy po wybraniu produktu wcisnąć klawisz „Mark". Ponowne wciśniecie klawisza „Mark" wycofa zaznaczenie. Zakupione produkty w kolumnie „Mark" posiadają wartość „true".

5.2.8 Odfiltrowanie zakupionych produktów

Produkty, które zostały już przez użytkownika zakupione mogą być automatycznie ukrywane. Uruchomienie opcji odfiltrowania produktów odbywa się przez wybranie opcji „Menu-> View-> Filtered" dostępnej z formatki prezentującej produkty z listy zakupów. Ponowne wybranie tej opcji anuluje filtrowanie.

5.2.9 Sortowanie

Domyślnie produkty na liście zakupów prezentowane są w kolejności ich dodawania. Użytkownik ma jednak możliwość wyboru sortowania elementów prezentowanych na liście zakupów po wybranej kolumnie.

Sortowanie uruchamiane jest przez wybranie z podmenu „Menu->View->Sort" kolumny, po której mają zostać posortowane elementy.

5.3 Baza artykułów

5.3.1 Podstawowe operacje na artykułach w bazie

Baza artykułów dostępna jest przy użyciu formatki o nazwie *Produkt Base,* do której można przejść z menu głównego aplikacji. Na bazie artykułów można wykonywać wszystkie podstawowe operacje podobnie jak na normalnej liście zakupów. Można dodawać, usuwać, edytować i sortować elementy w bazie. Dostęp do tych opcji odbywa się przez menu umieszczone pod prawym klawiszem szybkiego dostępu.

5.3.2 Pobieranie produktów do bazy ze sklepu internetowego

Oprócz ręcznego dodawania produktów do bazy istnieje też możliwość ściągnięcia listy dostępnych produktów z serwera sklepu internetowego. Wypełnienie bazy produktów informacjami ze sklepu umożliwia potem łatwe zamawianie produktów w tym sklepie, gdyż unika

się niejednoznaczności, która może powstać przy zamawianiu produktów zdefiniowanych przez użytkownika.

Aby pobrać listę produktów należy wybrać opcję „Menu->Actions->Fill”. Pojawi się okno dialogowe z informacjami o serwisie internetowymi, z którym ma nastąpić połączenie i z prośbą o potwierdzenie połączenia. Operacje trzeba zatwierdzić klawiszem „Yes” lub można zrezygnować i powrócić do poprzedniej formatki wciskając klawisz „No”.

Aplikacja łączy się z usługą sieciową `GetOfferedProducts` wybranego sklepu internetowego, która wysyła listę oferowanych produktów. Po otrzymaniu listy jest ona łączona z istniejąca bazą artykułów. W przypadku konfliktów informacje otrzymane z serwisu nadpisują istniejące pozycje w bazie.

5.3.3 Dynamiczne wyszukiwanie

Ze uwagi na potencjalnie dużą ilość produktów w bazie, bardzo ważne jest sprawne odnajdowanie pożądanych pozycji. Oprócz mechanizmu sortowania po dowolnej kolumnie aplikacja umożliwia również dynamiczne wyszukiwanie produktów po ich nazwie.

Funkcjonalność ta dostępna jest po wybraniu w menu opcji „Find”. Pojawia się wtedy dodatkowe pole, w którym należy wpisać nazwę poszukiwanego produktu. W trakcie pisania efekty wyszukiwania są widoczne w głównym oknie z listą produktów w bazie. Wystarczy wpisać kilka pierwszych liter szukanego artykułu, a następnie wybrać konkretną pozycję z wyświetlonej skróconej listy produktów. Dynamiczne wyszukiwanie prezentuje rysunek 5.5.

Rysunek 5.5: Dynamiczne wyszukiwanie

5.4 Współpraca z serwerem produktów

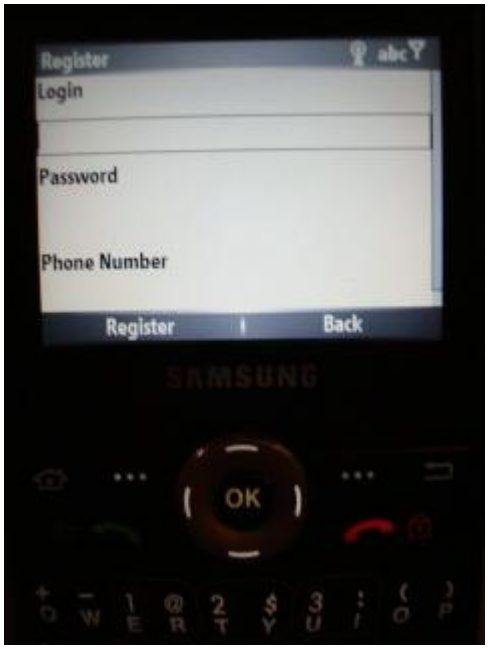
5.4.1 Wybór sklepu internetowego

Przed połączeniem się z usługami sieciowymi należy podać ich adres internetowy. Formatka *Connections* dostępna z menu głównego aplikacji umożliwia dodawanie i wybór sklepu, z którym chcemy się połączyć.

Aby dodać sklep należy wybrać z menu opcję „Add”. Istniejące wpisy można edytować korzystając z opcji „Edit”.

5.4.2 Rejestracja

Aplikacja mobilna umożliwia zamówienie produktów z list zakupów w sklepie z zainstalowanym serwerem produktów. Aby móc skorzystać z usługi zamawiania, należy najpierw zarejestrować się w serwisie. Można tego dokonać korzystając z formatki *Register* wywoływanej przez wybranie z menu formatki *Connections* opcji „Register”. W nowy oknie należy podać identyfikator, hasło i numer telefonu. Po podaniu danych i wciśnięciu klawisza „Register” wywoływany jest usługa sieciowa `RegisterUser` odpowiedzialna za stworzenie nowego użytkownika w bazie. Jeśli rejestracja przebiegnie pomyślnie, to użytkownik informowany jest o tym specjalnym komunikatem, a jeśli wystąpił błąd, to serwis przekazuje opis problemu. Po pomyślnej rejestracji można już zamawiać produkty w danym sklepie. Ekran rejestracji można zobaczyć na rysunku 5.6.



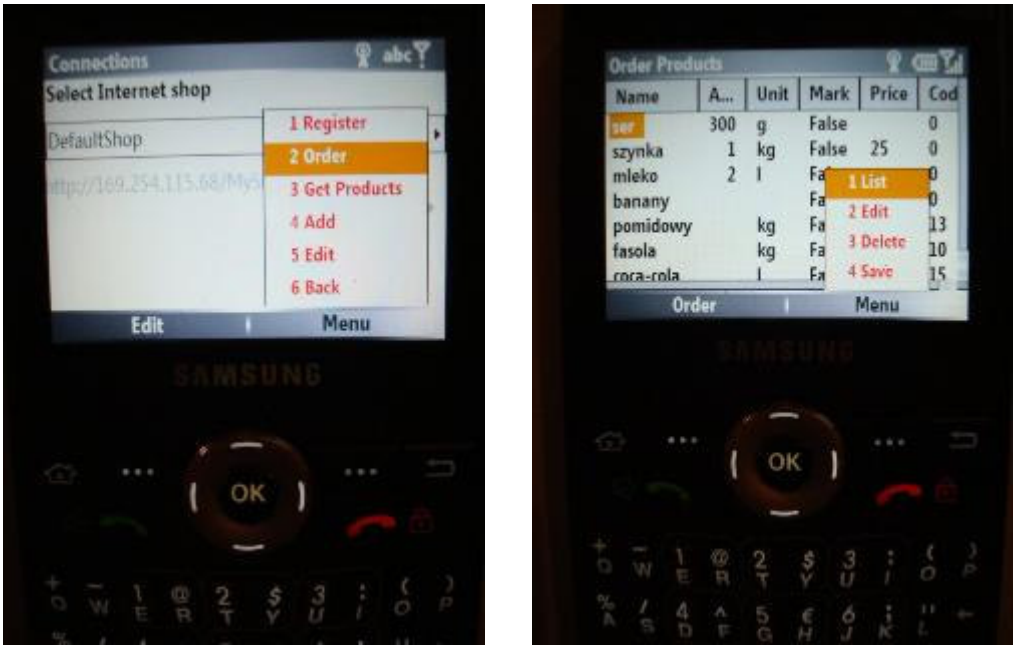
Rysunek 5.6: Rejestracja w sklepie internetowym

5.4.3 Zamawianie produktów z listy zakupów

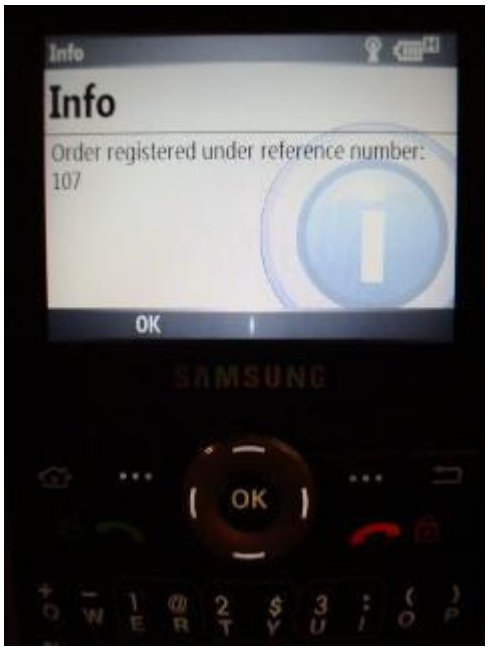
Zamawianie produktów odbywa się z poziomu formatki *Order Products*, którą wywołuje się z formatki *Connections* przez wybranie z menu opcji „Order”. Po wybraniu listy zakupów („Menu-> Lists”), którą chcemy zamówić należy wcisnąć klawisz „Order”. Program prosi o podanie identyfikatora i hasła do autentykacji użytkownika na serwerze produktów. Aplikacja ustanawia w tle połączenie z serwisem `PlaceOrder`, który jako argument pobiera `DataSet` zawierający tabelę z rekordami odpowiadającymi zamawianym produktom. Serwer porównuje otrzymane produkty z dostępnymi w systemie na podstawie pola „Code”. Produkty, które zostały dodane do listy zakupów na podstawie ściągniętej z serwera listy uaktualnień powinny mieć poprawnie wypełnione pole „Code”. Produkty, które zostały stworzone przez użytkownika ręcznie mają w polu „Code” wartość „0”. W takiej sytuacji serwer porównuje otrzymane nazwy artykułów z nazwami dostępnych produktów w sklepie. Jeśli uda się dopasować całą nazwę lub jej podciąg, to serwer dodaje taką pozycję w miejsce nadesłanej przez użytkownika. Jeśli choć jednej z pozycji na liście nie udało się jednoznacznie dopasować do artykułów z asortymentu sklepu, to usługa

sieciowa `PlaceOrder` przekazuje do aplikacji mobilnej `DataSet` z listą produktów uzupełnioną o sugerowane pozycje w miejsce niepasujących.

Użytkownik ma możliwość obejrzenia otrzymanej zwrotnej listy, może ją edytować i uzupełniać. Następnie ponownie wywołuje usługę `PlaceOrder`. Po poprawnym przyjęciu zamówienia przez serwis przekazywany jest numer zlecenia w systemie, na podstawie, którego można odebrać zamówienie. Kolejne etapy składania zamówienia ilustrują rysunki 5.7 i 5.8.



Rysunek 5.7: Procedura zamawiania produktów



Rysunek 5.8: Ekran informujący o przyjęciu zamówienia

Testy i koszty korzystania z aplikacji

5.5 Testy wydajnościowe

Do testów gotowej aplikacji wykorzystano smartphona firmy Samsung model SGH-i600. Telefon ma procesor Intel PXA27x z serii XScale (wykorzystujący technologię ARM), posiada 64MB pamięci RAM, 128MB ROM, 50MB wbudowanej pamięci flash na dane użytkownika oraz pamięć zewnętrzną w postaci karty SD o pojemności 1GB. Ekran telefonu pozwala wyświetlać obraz z rozdzielczością 320x240 w 16-bitowej palecie kolorów. Smartphone wyposażony jest w mobilną wersję klawiatury QWERTY.

Wydajność aplikacji przetestowano wczytując specjalnie przygotowane listy produktów zawierające odpowiednią liczbę elementów postaci:

```
<ItemList>
 <Name>testowy1</Name>
 <Amount>11</Amount>
 <Unit>g</Unit>
 <Mark>false</Mark>
 <Price>0,00</Price>
 <Code>0</Code>
</ItemList>
```

Czas mierzony był z wykorzystaniem klasy `DataTime`, której wartość była zapamiętywana przed wywołaniem formatki prezentującej produkty listy zakupów z listy testowej, a następnie porównywana z wartością `DateTime` po pełnym załadowaniu danych i wyświetleniu ich użytkownikowi. Niestety klasa `DateTime` w wersji na smartphona pozwala uzyskać tylko precyzję pomiaru z dokładnością do jednej sekundy (w wersji na komputery stacjonarne do jednej milisekundy).

Operacja	Czas
Odczyt i załadowanie do kontrolek listy zakupów zawierającej 100 rekordów	1-2 sekundy
Załadowanie do kontrolek listy zakupów zawierającej 100 rekordów z wykorzystaniem bufora	1 sekunda
Odczyt i załadowanie do kontrolek listy zakupów zawierającej 1000 rekordów	14 sekund
Załadowanie do kontrolek listy zakupów zawierającej 1000 rekordów z wykorzystaniem bufora	6-7 sekund

Tabela 2: Test wydajności aplikacji

Z przeprowadzonych testów wynika, że z aplikacją można w miarę wygodnie pracować z listami zakupów zawierającymi do paruset rekordów.

5.6 Koszty

Korzystanie z aplikacji wiąże się z kosztami wysyłania krótkich wiadomości tekstowych, oraz korzystania z transmisji danych GPRS do łączenia się z serwerem produktów. Przedstawione wyliczenia zostały wykonane na podstawie cennika operatora sieci komórkowej PlusGSM dla najniższej wersji abonamentu („Taryfa Kubali 25"). Tabela 2 zawiera koszt poszczególnych usług.

Usługa	Opłata
Opłata za SMS do dowolnego operatora sieci komórkowej	0,18 zł
Pakietowa Transmisja Danych (GPRS) Internet 100 KB (APN: www.plusgsm.pl, internet)	0,12 zł
Pakiet 200 SMS-ów (opłata miesięczna)	5zł
Minuta połączenia głosowego do dowolnego operatora sieci komórkowej	0,60gr

Tabela 2: Wybrane pozycje z cennika usług telefonicznych PlusGSM dla „Taryfy Kubali 25" z dnia 27.09.2007, http://www.plus.pl/oferta_indywidualna/plus_abonament/taryfy/taryfy_kubali

W celu oszacowania średniego kosztu korzystania z poszczególnych funkcjonalności przyjęto następujące założenia:

- Krótka wiadomość tekstowa może pomieścić maksymalnie 160 znaków.
- Nagłówek dołączany do każdego wysyłanego przez aplikację SMS-a ma średnio 30 znaków.
- Nazwy list i produktów mają średnio 15 znaków.
- Każda pozycja wysłana za pomocą SMS-a ma 28 znaków (15 znaków nazwa + 4 znaki ilość + 2 znaki jednostka + 3 znaki kod + 4 separatory). W przypadku, gdy pola nie wykorzystują założonego miejsca, liczba znaków będzie odpowiednio mniejsza.
- W jednym SMS-ie mieszczą się 4 pozycje (30 znaków nagłówek + 4 * 28 znaków na pozycje = 148 znaków).
- W rozliczeniu nie uwzględniono transmisji wymaganej do zainicjowania połączenia z operatorem i obsługi wykorzystanego protokołu (kilka kilobajtów)
- Zamówienia wymagające korekty wymagają 3 faz transmisji (wysłanie zamówienia, otrzymanie zestawu do korekty + ponowne wysłanie)
- Należy pamiętać, że operator nalicza opłatę za każde rozpoczęte 100KB danych.

Funkcjonalność	Rozmiar danych	Opłata
Wysyłanie listy zakupów z 20 pozycjami	20 / 4 = 5 SMS-ów	5 * 0,18 = 90gr
Ściągnięcie listy produktów z 300 elementami	76KB	~ 9gr
Wysłanie pustego zamówienia	2KB	~ 0 gr
Wysłanie zamówienia z 30 pozycjami	8KB	~ 1g
Wysłanie zamówienia z 30 pozycjami wymagającego korekty	8KB + 9KB + 8KB = 3 25KB	3gr

Tabela 3: Koszt wykorzystania poszczególnych funkcjonalności aplikacji

Z Tabeli 3 wynika, że koszt korzystania z usług pobierania listy produktów i składania zamówień jest znikomy. Wysyłanie list za pomocą, krótkich wiadomości tekstowych jest już znacznie droższe, ale ciągle wysłanie listy z 20 pozycjami odpowiada 1,5 minutowej rozmowie telefonicznej. Biorąc pod uwagę, że osoba otrzymująca taką listę w trakcie rozmowy głosowej musi ją gdzieś zapisywać, czas 90 sekund może okazać się niewystarczający. Przy częstym wykorzystaniu funkcjonalności wysyłania list za pośrednictwem SMS warto wykupić pakiet 200 SMS-ów za 5 zł. Koszt jednego SMS-a spada wtedy do 2,5gr.

Podsumowanie

Zaprezentowana aplikacja mobilna może być instalowana i z powodzeniem wykorzystywana na urządzeniach mobilnych typu smartphone, jak również nowych modelach PDA Phone. Do pełnego wykorzystania możliwości systemu należałoby znacząco rozbudować możliwości serwera produktów i przystosować go do kompleksowej obsługi zleceń. Można się jednak spodziewać, że sklep, który zdecydowałby się na wykorzystanie zaprezentowanego rozwiązania posiada już wymaganą infrastrukturę informatyczną do obsługi zamówień. Musiałby tylko przygotować odpowiednią wersję usług sieciowych wykorzystywanych przez aplikację mobilną do komunikacji z serwerem produktów. Dzięki zastosowaniu uniwersalnych usług sieciowych przygotowanie takiego rozszerzenia stosunkowo proste i możliwe niezależnie od technologii i platformy sprzętowej wykorzystywanej przez sklep.

Można też rozszerzać funkcjonalność samej aplikacji mobilnej. Możliwe usprawnienia to:

- Aplikacja konfiguracyjna – Działająca na komputerze stacjonarnym aplikacja umożliwiająca zmianę ustawień i list zakupów na podłączonym do niego urządzeniu mobilnym. Z przygotowania tej części projektu zrezygnowano ze względu na ograniczenia czasowe i jego stosunkowo małą użyteczność. Dostęp do plików XML z danymi aplikacji można uzyskać w bardzo prosty sposób wykorzystując darmowy program ActiveSync dołączony do każdego smartphona z Windows Mobile. Pliki XML można następnie zmodyfikować w dowolnym edytorze XML.

- Obsługa połączeń krótkiego zasięgu – Warto byłoby umożliwić użytkownikom przekazywanie list pomiędzy telefonami za pośrednictwem Bluetooth i Wi-fi. Umożliwiłoby to zaoszczędzenie pieniędzy na wysyłanie SMS-ów w przypadku, gdy użytkownicy znajdują się w jednym pomieszczeniu.

- Integracja z Pocket Outlook – Można rozszerzyć aplikację o mechanizm współpracy z wbudowanym w Windows Mobile kalendarzem i wykorzystać jego mechanizm przypominania i planowania czasu.

Na przykładzie zaprezentowanego programu do zarządzania listą zakupów przedstawiono nowy sposób myślenia i projektowania aplikacji mobilnych, gdzie główny nacisk kładzie się na zapewnienie użytkownikowi dostępu do potrzebnych mu informacji wtedy, kiedy są one mu najbardziej potrzebne wykorzystując do tego różne metody komunikacji i wymiany danych. Dla niektórych trzymanie się zasady dostępu do informacji „w dowolnym momencie, w dowolnym miejscu i z dowolnego urządzenia” już nie wystarcza. Niedługo możemy się spodziewać, że aplikacje mobilne będą naszym źródłem informacji „ zawsze, wszędzie i z każdego urządzenia”.

Dodatek A. Zawartość płyty CD

Na płycie CD dołączonej do pracy znajdują się:

- Praca magisterska w formacie PDF i DOC.
- W katalogu Source/ShoppingList znajdują się źródła aplikacji mobilnej.
- W katalogu Source/MyShop znajdują się źródła serwera produktów.
- W katalogu Instal znajdują się dodatkowe programy potrzebne przy instalacji serwera produktów.
- W katalogu ShoppingList znajduje się gotowy pakiet instalacyjny aplikacyjny aplikacji mobilnej.
- Opis instalacji znajduje się w pliku instal.txt.

Dodatek B. Pliki XML Schema Definition

W dodatku zamieszczone są pliki XML Schema Definition, na podstawie których tworzone są pliki XML przechowujące dane w aplikacji mobilnej.

Plik ListDataSet.xsd – na jego podstawie tworzony jest XML ze zbiorem wszystkich przechowywanych list produktów.

```
<?xml version="1.0" standalone="yes"?>
<xs:schema id="ListDataSet"
targetNamespace="http://tempuri.org/ListDataSet.xsd"
xmlns:mstns="http://tempuri.org/ListDataSet.xsd"
xmlns="http://tempuri.org/ListDataSet.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
attributeFormDefault="qualified"
elementFormDefault="qualified">
  <xs:element name="ListDataSet" msdata:IsDataSet="true"
  msdata:UseCurrentLocale="true">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ShopList">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string" />
              <xs:element name="listId" type="xs:string" />
              <xs:element name="smsNo" type="xs:int" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
    <xs:unique name="PrimeKey" msdata:PrimaryKey="true">
      <xs:selector xpath=".//mstns:ShopList" />
      <xs:field xpath="mstns:listId" />
    </xs:unique>
  </xs:element>
</xs:schema>
```

Plik ItemDataSet.xsd – na jego podstawie tworzone są pliki XML przechowujące produkty w ramach pojedynczej listy oraz lista dostępnych produktów ściągnięta z serwisu uaktualnień.

```
<?xml version="1.0" standalone="yes"?>
<xs:schema id="ItemDataSet"
targetNamespace="http://tempuri.org/ItemDataSet.xsd"
xmlns:mstns="http://tempuri.org/ItemDataSet.xsd"
xmlns="http://tempuri.org/ItemDataSet.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
attributeFormDefault="qualified"
elementFormDefault="qualified">
  <xs:element name="ItemDataSet" msdata:IsDataSet="true"
  msdata:Locale="en-US">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ItemList">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string" />
              <xs:element name="Amount" type="xs:string" />
              <xs:element name="Unit" type="xs:string" />
              <xs:element name="Mark" type="xs:boolean" />
              <xs:element name="Price" type="xs:string" />
              <xs:element name="Code" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
    <xs:unique name="NameKey1">
      <xs:selector xpath=".//mstns:ItemList" />
      <xs:field xpath="mstns:Name" />
    </xs:unique>
  </xs:element>
</xs:schema>
```

Bibliografia

[1] Architektura ARM, Wikipedia, http://en.wikipedia.org/wiki/ARM_architecture, 2007

[2] B'Far Reza, Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML, Cambridge University Press, 2005

[3] Cellular Network, Wikipedia, http://en.wikipedia.org/wiki/Cellular_network, 2007

[4] Christian Forsberg, Effective Memory, Storage, and Power Management in Windows Mobile 5.0, http://msdn2.microsoft.com/en-us/library/aa454885.aspx, 2006

[5] Denise Barnes, Fundamentals of Microsoft .NET Compact Framework Development for the Microsoft .NET Framework Developer, http://msdn2.microsoft.com/en-us/library/aa446549.aspx, 2003

[6] Embedded Linux Consortium, http://www.embedded-linux.org/, 2007

[7] Embedded Linux, Wikipedia, http://en.wikipedia.org/wiki/Embedded_Linux, 2007

[8] Garnet OS, http://www.access-company.com/products/garnet/index.html, 2007

[9] Ian Griffiths, BindableListView, http://www.interact-sw.co.uk/utilities/bindablelistview/source/, 2003

[10] Java Mobile Edition, http://java.sun.com/javame/index.jsp, 2007

[11] Microsoft Developer Network, http://msdn.microsoft.com/, 2007

[12] Moth Daniel, Foot Peter, Microsoft Mobile Development Handbook, Microsoft Press, 2007

[13] Palm, http://euro.palm.com/europe/, 2007

[14] PeiZheng, Lionel M. Ni, Smart Phone and Next-Generation Mobile Computing, Morgan Kaufmann Publishers, 2006

[15] Procesor Intel Xscale, Wikipedia, http://en.wikipedia.org/wiki/Intel_Xscale, 2007

[16] Symbian OS, http://www.symbian.com/, 2007

[17] Symbian OS v9.2 System Model, http://developer.symbian.com/main/downloads/files/ 2007

[18] Technologia Xscale, http://www.intel.com/design/intelxscale/, 2007

[19] Transmeta Corporation, http://www.transmeta.com/tech/ 2007

[20] Wei-Meng Lee, Design Considerations for Microsoft Smartphone Applications, http://www.oreillynet.com/pub/a/wireless/2004/01/07/smartphone.html, 2004

[21] Windows Mobile 5.0 SDK Documentation, http://msdn2.microsoft.com/en-us/library/ms880496.aspx, 2005

[22] Windows Mobile Developer Center, http://msdn.microsoft.com/windowsmobile/, 2007

[23] Windows Mobile http://www.microsoft.com/poland/windowsmobile/default.mspx, 2007