

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Informatica

**STUDIO E SVILUPPO DI
UN'APPLICAZIONE MOBILE PER iOS
NELL'AMBITO DELLA BI:
UN TASK MANAGER PER SAP
BUSINESS PLANNING AND
CONSOLIDATION**

Tesi di Laurea in Complementi di Basi di Dati

Relatore:
Chiar.mo Prof.
DANILO MONTESI

Presentata da:
GIULIA ALBANESE

Correlatore:
Ing.
MARCO SCHONHAUT

**Prima sessione
2011/2012**

*Alle persone più importanti della mia vita senza le quali non ce
l'avrei mai fatta!*

Introduzione

Ci sono delle attività che devono essere eseguite in un determinato momento e alcune volte non si sa quale sia questo momento. Non è quindi possibile né eseguire né schedare l'esecuzione di queste attività. Se ad esempio alla fine di un'attività deve partirne un'altra ma non si sa quando la prima terminerà, è necessario attenderne il completamento prima di poter eseguire la successiva. Può, inoltre, capitare che le attività da fare risultino così urgenti da doverci dedicare ore extra fuori dall'orario lavorativo con la necessità, la maggior parte delle volte, di essere sul posto di lavoro o di doversi portare dietro il portatile aziendale anche in vacanza. In questi casi sarebbe comodo poter utilizzare un dispositivo di più facile trasporto e meno ingombrante che contenga gli strumenti necessari all'esecuzione di queste attività lavorative. Quale dispositivo, se non un tablet o uno smartphone, si presta meglio ad essere trasportato senza sentirne il peso?

Mentre esistono già soluzioni mobili per la fruizione di report di analisi avanzata, alla data attuale lo strumento della suite SAP BPC¹ non dispone ancora di moduli o client dedicati ai dispositivi mobili.

In particolare i processi di Pianificazione e Budget di grandi realtà necessitano di tempi strettissimi per la produzione degli scenari previsionali e il Process Owner, che coordina i tanti attori coinvolti nel processo, deve procedere lungo gli step di determinazione della pianificazione delle voci di conto economico con estrema tempestività onde non risultare collo di bottiglia del processo stesso. Essendo tale figura spesso chiamata a intervenire in riunioni o fuori

¹**BPC**: Business Planning and Consolidation

sede, risulterebbe di grande utilità dotare il software di pianificazione di un'applicazione che consenta di eseguire gli step di processo nella produzione del budget anche da sistemi mobili. Attualmente l'interfaccia di SAP BPC è gestita interamente su un client Microsoft Excel tramite un add-in che viene richiamato da un workflow pubblicato via web.

Scopo di questo lavoro di tesi è uno studio sullo stato dell'arte delle strategie adottate da SAP in questo ultimo periodo e la progettazione e realizzazione di **iTwin BPC**, un task manager per SAP BPC per Microsoft, ovvero uno strumento che permetta di programmare l'esecuzione e visualizzare lo stato dei DTS² da dispositivi mobili. Questa applicazione è sviluppata in Twinergy S.p.A.³ dove lavoro come *BI Consultant* da più di un anno.

Il presente documento è organizzato secondo il seguente schema: nel **Capitolo 1** viene fatto uno studio sugli strumenti mobili e l'influenza che hanno avuto nella Business Intelligence. Nel **Capitolo 2** viene presentata la strategia SAP del 2011-2012 riguardante la mobilità. Nel **Capitolo 3** viene introdotto iTwin BPC. Nel **Capitolo 4** ne viene mostrata l'architettura. Nei **Capitoli 5 e 6** vengono descritti rispettivamente la parte server side e la parte client side dell'applicazione. Infine vengono presentati possibili sviluppi futuri per iTwin BPC. Sono presenti, inoltre, le **Appendici A e B** nelle quali vengono approfondite alcune tecnologie utilizzate.

²DTS: Data Transformation Services. Procedure di elaborazione dati

³Twinergy S.p.A.: società di consulenza per la progettazione e attivazione di sistemi informativi e specializzata nell'implementazione di strumenti a supporto delle decisioni e del management aziendale. www.twinergy.com

Indice

Introduzione	i
1 Gli strumenti mobili	1
1.1 Ruolo nelle applicazioni di business	1
1.2 Evoluzione del mercato per tablet e smartphone	2
1.3 Principali utilizzi in ambito di lavoro e la loro evoluzione	3
1.4 Ruolo nella BI	3
1.5 Tre tendenze per il futuro	4
2 Integrazione della Suite SAP con strumenti mobile	5
2.1 La strategia 2011-2012 della SAP	5
2.1.1 In-Memory	7
2.1.2 Mobility	9
2.2 SAP Store	10
3 Progettazione di iTwin BPC	13
3.1 Introduzione a SAP BPC	13
3.1.1 Struttura architetturale	14
3.2 Specifiche del progetto	16
3.2.1 Requisiti funzionali	17
3.2.2 Requisiti non funzionali	18
3.2.3 Casi d'uso	19

4	Architettura del task manager	27
4.1	JDBC per SQL Server	29
4.2	Server side (Tier 2)	29
4.3	Server side (Tier 1)	30
4.4	Client side (Tier 0)	41
5	Sviluppo applicazione server side	43
5.1	Web Service	43
5.1.1	Diagramma delle classi	43
5.2	Presentazione web	48
5.3	Diagrammi di sequenza	48
5.3.1	Esecuzione di un DTS	49
5.3.2	Schedulazione di un DTS	49
5.3.3	View status dei DTS	49
6	Sviluppo interfaccia client-side	53
6.1	Paradigma di sviluppo	54
6.2	Tecnologie utilizzate	54
6.3	Diagramma delle classi	55
6.3.1	Model	56
6.3.2	ViewControllers	58
	Conclusioni e sviluppi futuri	73
A	Web Service	75
A.1	WSDL	77
A.2	SOAP	77
A.3	UDDI	77
B	iOS	79
B.1	iOS SDK	79
B.2	Storyboards	80
B.3	App Store	81

B.4 Protocols e Delegation	82
Bibliografia	85

Elenco delle figure

3.1	Schema architetturale SAP BPC su singola macchina	15
3.2	Schema architetturale di SAP BPC distribuito su più macchine	16
3.3	Nomenclatura del database di SAP BPC	17
3.4	Diagramma utenti	19
3.5	Diagramma di attività: autenticazione	21
3.6	Diagramma di attività: esecuzione di un package	22
3.7	Diagramma di attività: visualizzazione dello stato dei package	23
3.8	Diagramma di attività: schedulazione di un package	25
4.1	Schema architetturale	28
4.2	Schema architetturale in dettaglio tra tier 0 e tier 1	28
5.1	Diagramma delle classi del web service	44
5.2	Diagramma di sequenza: esecuzione di un DTS	50
5.3	Diagramma di sequenza: schedulazione di un DTS	51
5.4	Diagramma di sequenza: view status dei DTS	52
6.1	Diagramma delle classi dell'applicazione iOS (Model)	55
6.2	Diagramma delle classi dell'applicazione iOS (ViewControllers)	56
6.3	Screenshot applicazione: login	60
6.4	Screenshot applicazione: lista connessioni salvate	61
6.5	Screenshot applicazione: lista environment	62
6.6	Screenshot applicazione: lista model	63
6.7	Screenshot applicazione: menù	64

6.8	Screenshot applicazione: lista DTS	65
6.9	Screenshot applicazione: lista stati DTS	66
6.10	Screenshot applicazione: lista stati DTS con ordinamento e in dettaglio	67
6.11	Screenshot applicazione: lista parametri DTS	68
6.12	Screenshot applicazione: scelta singola parametri	69
6.13	Screenshot applicazione: scelta multipla parametri	70
6.14	Screenshot applicazione: menu run/schedule	71
6.15	Screenshot applicazione: esecuzione e schedulazione DTS . . .	72
A.1	Interazione tra Client, Web Service e Registry	76
B.1	Storyboard	81

Elenco dei listati

4.1	Documento XML: index.jsp (new)	30
4.2	Documento XML: index.jsp (old)	30
4.3	Documento XML: index.jsp (error)	31
4.4	Documento XML: login.jsp	32
4.5	Documento XML: login.jsp (error)	32
4.6	Documento XML: logout.jsp	32
4.7	Documento XML: environment.jsp	33
4.8	Documento XML: model.jsp	33
4.9	Documento XML: menu.jsp	34
4.10	Documento XML: dtsList.jsp	35
4.11	Documento XML: dtsParam.jsp	36
4.12	Documento XML: dtsStatus.jsp	38
4.13	Documento XML: dtsSchedule.jsp	39
4.14	Documento XML: dtsRun.jsp	40

Capitolo 1

Gli strumenti mobili

Nello scenario lavorativo odierno l'utilizzo degli strumenti mobili è ormai una componente insostituibile. Lo sviluppo di nuovi hardware e sistemi software ha modificato il modo di comunicare, rendendo tutto più facile e veloce.

1.1 Ruolo nelle applicazioni di business

L'uso sempre più pervasivo di tablet e smartphone sta trasformando anche le applicazioni di Business Intelligence, Analytics e Performance Management.

Le aziende hanno tutta l'intenzione di cavalcare il fenomeno, con l'obiettivo di velocizzare e rendere più semplice l'accesso alle informazioni di business. Sono queste le prime conclusioni di un'indagine condotta a metà 2011 da Axiante¹ sui responsabili IT di 200 imprese medio-grandi attive in Italia nei settori manifatturiero, commercio e servizi. Obiettivo della ricerca è quello di tastare il polso agli ICT manager sulle nuove possibilità offerte dai trend emergenti nel mondo "mobile". Tra le importanti conclusioni emerse dall'indagine è da sottolineare che il tema della Business Intelligence in mo-

¹**Axiante**: società di consulenza applicativa che lavora sia in Italia che a livello internazionale. www.axiante.com

bilità attira con prepotenza l'attenzione: se solo due anni fa, nel 2009, la mobile BI era considerata futuristica e nel 2010 era diventata un argomento di curiosità, oltre il 70% delle aziende coinvolte nell'indagine effettuata da Axiante afferma che attualmente sta pianificando la sua realizzazione. Dal punto di vista degli utenti, il ruolo fondamentale in questa evoluzione sarà svolto dai manager, che dovrebbero trascinare tutto il resto dell'organizzazione in un processo di tipo top-down. Sono infatti in molti a ritenere che l'utilizzo dei dispositivi mobili per le funzioni di BI potrebbe soppiantare l'uso del PC in un futuro molto prossimo. Da un punto di vista funzionale, uno spazio sempre maggiore sarà occupato dalle "viste" di tipo grafico, che sono adatte a essere visualizzate sui dispositivi tablet, e alle rappresentazioni sintetiche, perfette per gli smartphone.

1.2 Evoluzione del mercato per tablet e smartphone

Proprio quando si pensava di aver finalmente sviluppato interfacce tra il sistema di business intelligence e i desktop dei clienti, tra i computer portatili e gli smartphone, compare un'altra categoria di dispositivi: i tablet. Questi dispositivi, a differenza degli smartphone di prima generazione, hanno capacità maggiori per quanto riguarda l'accesso al sistema di dati di business intelligence e strumenti analitici. Gli smartphone sono dispositivi ideali per fornire avvisi, visualizzare importanti indicatori di prestazione e file PDF di anteprima dei report per i dirigenti senior e per i "guerrieri della strada". Comunque, le dimensioni ridotte dello schermo e i tasti piccoli precludono un utilizzo di tipo analitico, indagini *drill down*² e altri esercizi BI standard.

²**Drill down:** capacità di penetrare nei dati a diversi livelli di gerarchie

1.3 Principali utilizzi in ambito di lavoro e la loro evoluzione

Con gli strumenti mobili l'utente business potrà avere accesso sempre e ovunque ad analisi e dati "in diretta".

A differenza delle applicazioni BI mobili di prima generazione, che consentivano di consultare solamente report statici pre-pacchettizzati con visualizzazioni limitate, adesso le nuove applicazioni mobili cercano di permettere agli utenti di comprendere quanto accade in tempo reale.

Secondo quanto riportato da Gartner³: "entro il 2013, il 33% delle funzionalità della business intelligence verrà consumato tramite dispositivi portatili". Con la forza lavoro mobile in costante aumento e lo scenario dei dispositivi mobili in continua evoluzione, diventa sempre più indispensabile mettere gli utenti nelle condizioni di poter analizzare le informazioni a 360°, effettuare qualunque associazione desiderino, e personalizzare le applicazioni secondo le esigenze individuali.

1.4 Ruolo nella BI

Business Intelligence mobile (Mobile BI o Mobile Intelligence) si riferisce alla distribuzione dei dati aziendali per dispositivi mobili quali smartphone e tablet PC. Business Intelligence (BI) si riferisce alle tecniche di computer-based utilizzate in *spotting*, *digging-out* e analisi dei dati aziendali, come il fatturato dei prodotti e/o dipartimenti o dei relativi costi e redditi.[1]

Secondo Aberdeen Group⁴, un gran numero di aziende vengono rapidamente impegnate con la BI mobile a causa del gran numero di pressioni del mercato, come la necessità di una maggiore efficienza nei processi aziendali,

³**Gartner Inc.:** società multinazionale leader mondiale nella consulenza strategica, ricerca e analisi nel campo dell'Information Technology. www.gartner.com

⁴**Aberdeen Group:** fornitore di ricerche basate sui fatti per aiutare le organizzazioni e i singoli individui a prendere le migliori decisioni di business.

il miglioramento della produttività dei dipendenti (ad esempio, il tempo trascorso alla ricerca di informazioni), migliorare e velocizzare il processo decisionale, migliorare il servizio al cliente e la consegna in tempo reale dell'accesso bidirezionale ai dati per poter prendere decisioni sempre e ovunque.[2]

1.5 Tre tendenze per il futuro

I dispositivi abilitati alle app sono da per tutto. I dispositivi “non-PC” non stanno sostituendo i PC ma si stanno comunque espandendo nel mercato.

Le imprese stanno “abbracciando” i tablet. Il COO⁵ della Apple ha affermato che il 65% delle aziende o stanno distribuendo ipad oppure sono in fase di sperimentazione di un analogo progetto. Storicamente, invece, le aziende sono molto più “lente” per quanto riguarda l'adozione di nuove strategie.

La maggior parte delle aziende vedono l'iPad come un sostituto al portatile, ma l'impatto reale avverrà attraverso la sostituzione degli appunti cartacei o come strumento di tecnologia complementare che non era disponibile prima.

Cambiamento della realizzazione del Business. Le aziende stanno usando i tablet non solo per leggere le email e per navigare in internet ma anche per le presentazioni ai clienti e per il supporto alle vendite.

⁵COO: Chief Operating Officer

Capitolo 2

Integrazione della Suite SAP con strumenti mobile

In questo periodo Sap e Sybase stanno trasformando il modo di lavorare delle imprese e delle persone[4]. Grazie all'acquisizione di Sybase, Sap sta cercando di dare la possibilità ai suoi clienti di sfruttare la potenza della BI in qualsiasi posto e da qualsiasi dispositivo. Sap e Sybase presentano una più ampia scelta di soluzioni per l'Enterprise Information Management (EIM), attraverso il porting, la certificazione e l'ottimizzazione di SAP Business Suite e altre soluzioni sui data management servers di Sybase, garantendo così ai clienti una scelta più ampia di piattaforme attraverso cui poter utilizzare le applicazioni Sap. Infine, per le aziende che utilizzano BI, Sap e Sybase offrono ai clienti un'infrastruttura di business analytics altamente performante che sfrutta le soluzioni BI di Sap sui server di data management; in questo modo è possibile, per i clienti, reperire informazioni attraverso data warehouse, query reporting, scorecarding, dashboarding, strumenti OLAP e mobile BI.

2.1 La strategia 2011-2012 della SAP

Sono due gli elementi cardine della nuova strategia SAP volta a supportare le aziende nel loro processo di crescita attraverso soluzioni di real time: l'**in-**

memory e la **mobility**.^[5]

Con il termine in-memory database (IMDB) si intende un DBMS che gestisce i dati nella memoria centrale. Esso è in contrasto con i DBMS classici che mantengono invece i dati su memorie di massa (dischi rigidi). Tenere i dati in memoria centrale permette tempi di elaborazione inferiori rispetto che tenerli su memorie di massa.

Con mobility, invece, viene indicata una strategia che tende ad estendere i servizi aziendali fuori dalle mura, permettendone l'utilizzo attraverso dispositivi mobili di ultima generazione (tablet e smartphone) dovunque essi si trovino.

In un momento storico in cui i ritmi e la velocità della società attuale impattano significativamente sulle dinamiche di mercato, l'innovazione tecnologica può fare la differenza nell'accelerazione del business.

La strategia di SAP sta evolvendo in linea con lo scenario ICT in cui si assiste al passaggio dal mondo strutturato dei processi al mondo destrutturato delle applicazioni in mobilità.

La consumerizzazione¹ dell'IT e la pervasività dei dispositivi mobili trasferiscono elementi come l'usabilità e la portabilità anche al mondo aziendale, dove la richiesta di applicazioni gestionali di nuova generazione e di soluzioni di Business Intelligence da fruire in mobilità sta aumentando considerevolmente.

La strategia SAP consente di affrontare progetti articolati in tempi brevi impattando in modo significativo su produttività ed efficienza e abilitando gli utenti di business ad essere sempre connessi con il patrimonio informativo aziendale attraverso le modalità semplici e immediate che utilizzano per comunicare nel tempo libero.

¹**Consumerizzazione:** fenomeno iniziato negli ultimi anni, con l'arrivo di ultimi dispositivi come smartphone e tablet, che sta cancellando il confine tra lavoro e vita privata dando la possibilità di scegliere come e dove lavorare e quali tecnologie utilizzare.

2.1.1 In-Memory

Il volume di dati che le organizzazioni devono gestire è sempre più elevato, e i tempi per prendere le decisioni importanti di business sono sempre più brevi.

Sul fronte in-memory computing, SAP intende rispondere all'information overflow che determina la proliferazione a una velocità esponenziale dei dati rendendo sempre più difficile interpretarli e trasformarli in informazioni utili per la strategia aziendale. Grazie al motore analitico in-memory SAP HANA² è possibile affrontare questa complessità informativa per analizzare enormi quantità di dati eterogenei in modo rapido e flessibile, presentandoli in forma semplice a qualunque utente ne abbia bisogno per prendere velocemente decisioni strategiche. Questa appliance multifunzionale, che abbina componenti software SAP ottimizzati per l'hardware di partner SAP, abilita applicazioni analitiche specifiche che consentono alle aziende di analizzare elevati volumi di informazioni al loro verificarsi, offrendo una visione sempre aggiornata dell'andamento del business e migliorando processi complessi come previsione e pianificazione.

Con SAP In-Memory Computing si ottengono rivoluzionarie innovazioni hardware e software, che consentono di gestire e analizzare consistenti volumi di dati in modo immediato ed efficace, riducendo al tempo stesso la complessità del sistema informatico.[7]

SAP HANA

Il motore di in-memory computing SAP, al centro di SAP HANA, è uno strumento integrato di calcolo e database che consente di elaborare grandi quantità di dati in tempo reale nella memoria centrale, per fornire istantaneamente i risultati di analisi e transazioni. Al pari di un comune database, il motore di in-memory computing SAP supporta gli standard di settore, come SQL e MDX, ma include anche un motore per il calcolo ad alte prestazioni

²SAP HANA: SAP High-Performance Analytic Appliance

che integra il supporto per il linguaggio procedurale direttamente nel kernel del database. Tale approccio elimina l'esigenza di leggere i dati dal database, elaborarli e quindi riscriverli nuovamente nello stesso.[8]

SAP presenta attraverso applicazioni nuove e potenziate, una gestione dei dati di nuova generazione e analisi innovative:

- **Trasformare il modo in cui le persone pensano, pianificano e lavorano con Applications Advanced di SAP In-Memory Computing.** SAP fornisce una nuova classe di applicazioni aziendali prima inimmaginabili, che combinano transazioni di volume elevato con analisi in tempo reale per migliorare drasticamente i processi ricchi di dati come la pianificazione, la previsione e l'ottimizzazione dei prezzi.
- **Sfruttare le possibilità dei dati in tempo reale con SAP In-Memory Database.** Integrando i progressi nella compressione dei dati, nell'archiviazione dei dati a colonne e nella tecnologia in-memory, SAP In-Memory Database è progettato per potenziare la nuova generazione di gestione dei dati aziendali.
- **Ottenere conoscenze più approfondite con le innovative applicazioni per analisi aziendale implementate in SAP In-Memory Appliance (SAP HANA).** SAP HANA è un'appliance in-memory flessibile, multifunzionale e compatibile con qualsiasi fonte di dati, abina componenti software SAP ottimizzati all'hardware fornito e distribuito dai principali hardware Partner di SAP. Con SAP HANA, le organizzazioni possono analizzare operazioni di business basate su elevati volumi di informazioni dettagliate man mano che queste si sviluppano.

SAP HANA, quindi, consente alle aziende di analizzare informazioni di business basate su elevati volumi di dati di dettaglio man mano che questi si sviluppano. Le organizzazioni possono esplorare ed analizzare immediatamente tutti i loro dati transazionali e analitici da praticamente qualsiasi fonte di dati in tempo reale. I dati operativi vengono acquisiti nella memoria

mentre si verifica una determinata transazione e, grazie alle viste flessibili, le informazioni analitiche sono accessibili alla velocità del pensiero. I dati esterni possono essere aggiunti ai modelli analitici per estendere l'analisi in tutta l'organizzazione.

2.1.2 Mobility

In particolare sul fronte mobility SAP mira a disaccoppiare la componente di gestione dei device da quella applicativa. Tradizionalmente device, applicazioni, sicurezza e policy aziendali rappresentano una catena di elementi guidati principalmente dai dispositivi, i cui i produttori tendono a imporre i propri standard, determinando diseconomie per i dipartimenti IT, costretti a gestire standard diversi a livello hardware e applicativo. Con le proprie soluzioni frutto dell'integrazione con Sybase, SAP intende invece creare un format standard di sviluppo perfettamente interoperabile. Una piattaforma unica riduce il TCO³ e semplifica la gestione delle problematiche di sicurezza e policy permettendo a device di diversi produttori di accedere alla medesima piattaforma applicativa.

SAP Sybase Unwired Platform

Basata su una tecnologia leader del settore, Sybase Unwired Platform è una piattaforma di applicazioni mobili che consente di sviluppare in modo semplice e rapido applicazioni mobili che offrono agli utenti aziendali un accesso sicuro a diversi dati aziendali su un'ampia gamma di dispositivi mobili.[6]

Con Sybase Unwired Platform si potrà:

- **Accelerare lo sviluppo di applicazioni mobili.** Un potente ambiente di tool 4GL consente di accelerare lo sviluppo di applicazioni.

³**Total Cost of Ownership:** costo totale di un computer o di un altro dispositivo. Oltre all'acquisto, si calcola anche l'installazione, i dispositivi accessori ed indispensabili, la manutenzione, i ricambi programmati ed ogni altro costo necessario al funzionamento corretto.

La compatibilità con i comuni ambienti di sviluppo integrati (IDE), quali Eclipse, consente agli sviluppatori di sfruttare l'expertise e gli strumenti preesistenti.

- **Supportare svariate tipologie di dispositivi mobili.** La piattaforma offre la capacità di un'unica progettazione e implementazione su diversi dispositivi mobili e sistemi operativi, tra cui Windows Mobile, Windows 32 (laptop/tablet), iPhone, Android e dispositivi RIM BlackBerry.
- **Estendere i dati dei sistemi di back-end ai dispositivi mobili.** La piattaforma si integra con numerose applicazioni enterprise, inclusa qualsiasi applicazione che sfrutta database o servizi SOA (Service-Oriented Architecture)/Web.
- **Implementare una soluzione per la sicurezza e la gestione della mobilità.** La perfetta integrazione con la soluzione Afaria per la sicurezza e la gestione di dispositivi mobili permette di utilizzare un'unica console amministrativa per la gestione, la sicurezza e l'implementazione centralizzata di dati, applicazioni e dispositivi mobili.
- **Avere la garanzia di una soluzione flessibile nel tempo.** Un'infrastruttura aperta e flessibile ti consente di rispondere in modo strategico all'evoluzione delle tipologie di dispositivi e delle fonti di dati all'interno dell'azienda.

2.2 SAP Store

Il 20 dicembre 2011 SAP AG e Sybase, Inc. hanno annunciato una nuova serie di applicazioni mobili progettate per una vasta gamma di settori e ruoli aziendali che operano in mobilità, dagli executive, ai knowledge worker, agli utenti operativi.[9] Grazie a queste nuove applicazioni e alla piattaforma mobile messa a disposizione, oltre ad espandere il proprio portfolio

mobile, unico nel settore, SAP rende possibile il reale superamento del *digital divide*, portando in “palmo di mano” la gestione della propria azienda, indipendentemente dal device utilizzato.

Le nuove applicazioni rendono fruibili in mobilità funzionalità di business strategiche, permettendo ai dipendenti aziendali di lavorare in qualunque momento e da qualsiasi luogo, utilizzando qualsiasi dispositivo mobile. Lo sviluppo di nuove apps è un’ulteriore dimostrazione della rapidità d’innovazione di SAP e della sua capacità di soddisfare la domanda degli utenti business, che si aspettano ormai un’esperienza mobile identica a quella user-friendly delle applicazioni consumer.

Per consentire ai clienti di accedere e scaricare rapidamente le nuove apps mobili, SAP ha lanciato SAP Store⁴, un “negozi online” dedicato alle aziende, che offre la stessa semplicità d’uso di un *app store* di tipo consumer, ma è progettato su misura per le esigenze dei clienti aziendali. SAP Store è disponibile sia come sito web, sia nella versione per dispositivi iOS come SAP Mobile Apps (a questa seguiranno le versioni per Android, BlackBerry e altre piattaforme).

Su SAP Store sono disponibili sia le applicazioni sviluppate da SAP, sia quelle certificate e sviluppate dal suo ampio ecosistema di Partner. I clienti potranno scaricare in prova le applicazioni prima dell’acquisto e richiedere quotazioni. SAP prevede di offrire presto ai clienti la capacità di acquistare direttamente le applicazioni su SAP Store e renderne sempre più semplice la distribuzione e implementazione negli ambienti aziendali.

⁴SAP Store: <https://store.sap.com/>

Capitolo 3

Progettazione di iTwin BPC

iTwin BPC è un'app add-in di SAP BPC che consente di estendere ai device mobili alcune funzionalità della suite SAP per la pianificazione e per il bilancio consolidato, in particolare le funzionalità che iTwin BPC al momento esporta sono:

- il check dello stato di esecuzione dei DTS;
- la possibilità di lanciare o schedulare DTS.

Per la progettazione dell'applicazione si è tenuto conto delle specifiche necessità che hanno espresso gli utenti che utilizzano SAP BPC.

3.1 Introduzione a SAP BPC

SAP BPC è uno strumento per la gestione delle performance aziendali facente parte della suite EPM¹. Esso consente, tramite strumenti di Business Intelligence e strutture dati multidimensionali, la creazione di scenari di pianificazione e previsionali di business.

SAP BPC consente la gestione e la misurazione delle performance aziendali tramite processi strutturati definibili dagli utenti di business.[3]
Lo strumento integra le funzionalità di Microsoft Office trasformandolo da

¹**EPM**: Enterprise Performance Management

strumento di lavoro individuale a strumento di lavoro condiviso su web e fornendo comunque all'utente un ambiente familiare, intuitivo e di immediata accettazione, in grado di ottimizzare i processi di business.

Le aziende si affidano a SAP BPC per soddisfare esigenze quali ad esempio:

- Pianificazione Budgeting
- Forecast, simulazioni, analisi what if
- Consolidato civilistico e gestionale
- Data entry manuale ed automatico di file
- Cruscotti direzionali
- Reporting operativo e business analysis

A questo scopo sono predisposte funzionalità per l'analisi degli scostamenti e l'individuazione delle principali cause delle variazioni. È prevista inoltre la massima flessibilità nella definizione delle dimensioni di analisi, nella generazione della reportistica e nelle schede di data entry.

Principali benefici:

- **Semplicità:** accedere alle giuste informazioni che servono in modo semplice
- **Agilità:** rendere il business capace di tenere il passo del cambiamento
- **Integrazione:** avere una architettura tecnologica flessibile in grado di supportare ogni tipo di bisogno riducendo i costi di gestione

3.1.1 Struttura architetturale

L'architettura di SAP BPC per Microsoft si compone di 3 tier:

- **Web Tier:** interfaccia utente per BPC web;

- **Application Tier:** servizi XML, logica di Business, linguaggio delle query condivise;
- **Database Tier:** archivio di dati relazionale e multi-dimensionale.

Questi tre tier possono essere inseriti in una sola macchina o distribuiti su più macchine, in base alle necessità aziendali. In particolare nel caso in cui si stia configurando un ambiente di sviluppo e il numero di utenti simultanei sia basso; in caso non ci siano requisiti per separare la componente MS IIS dalla componente Database SQL e non siano richiesti requisiti di alta disponibilità (clustering) allora si può pensare di utilizzare una sola macchina che contenga tutti e tre i livelli architetturali.

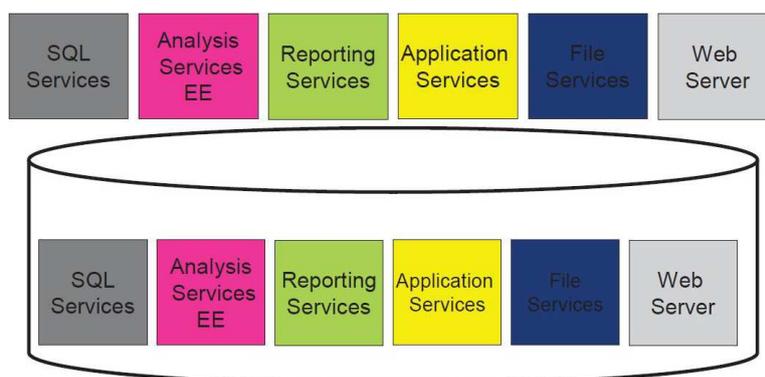


Figura 3.1: Schema architetturale SAP BPC su singola macchina

Nel caso in cui, invece, ci fossero politiche aziendali per cui sia necessario separare alcuni componenti (ad esempio IIS e SQL non debbano condividere lo stesso server); in caso il numero di utenti simultanei sia elevato e si abbia la necessità di separare i componenti funzionali in base alla struttura dipartimentale aziendale allora si deve pensare a una distribuzione dei livelli architetturali su più macchine.

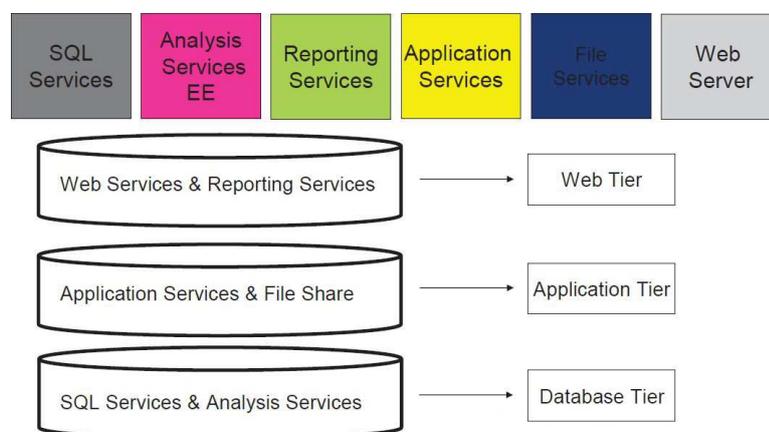


Figura 3.2: Schema architetturale di SAP BPC distribuito su più macchine

Per quanto riguarda l'autenticazione, BPC utilizza:

- tra Client e Web Server e tra Client e Application Server viene utilizzata l'autenticazione di windows o di SAP BO;
- tra Application Server e BPC System Resources (SQL services, Analysis services, File Services) Application Server utilizza il servizio a livello di account per accedere alle risorse.

Introduciamo la nomenclatura di SAP BPC per poter essere facilitati nei paragrafi successivi di questa tesi. Con **Environment** si intende un Data Warehouse contenuto all'interno del server, mentre per **Model** intendiamo un cubo di analisi all'interno del Data Warehouse. Di seguito viene mostrato un grafico che mostra la relazione tra environment e model.

3.2 Specifiche del progetto

Il progetto riguarda la creazione del modulo software di gestione dei processi di pianificazione per SAP BPC su mondo Microsoft. Questo software dovrà essere in grado di autenticare un utente, dare la possibilità di eseguire un DTS istantaneamente oppure di schedarne l'esecuzione, infine l'utente

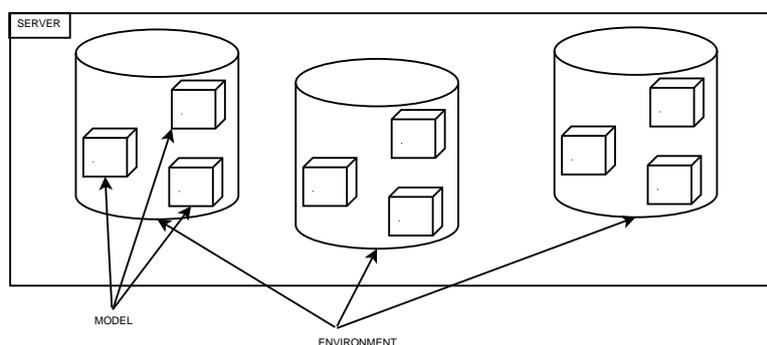


Figura 3.3: Nomenclatura del database di SAP BPC

dovrà essere in grado di controllare lo stato dei DTS che sono stati eseguiti a sistema.

3.2.1 Requisiti funzionali

Per ogni requisito vengono forniti un nome mnemonico, una breve descrizione e un codice di priorità prestabilito, nell'ottica di definire una gerarchia tra le funzionalità che dovranno essere implementate nel progetto.

- **Requisito #1**

- **Nome:** Autenticazione dell'utente
- **Descrizione:** L'utente deve potersi autenticare al sistema che deve riconoscerlo mostrandogli il *working set* di competenza
- **Rilevanza:** Essenziale

- **Requisito #2**

- **Nome:** Esecuzione di un package
- **Descrizione:** L'utente deve poter scegliere un package da eseguire e inserire i relativi parametri di input necessari
- **Rilevanza:** Essenziale

- **Requisito #3**

- **Nome:** Visualizzazione stato dei package eseguiti
- **Descrizione:** L'utente deve poter visualizzare lo stato di avanzamento dei package eseguiti (a livello di model)
- **Rilevanza:** Essenziale

- **Requisito #4**

- **Nome:** Schedulazione di un package
- **Descrizione:** L'utente deve poter scegliere un package da eseguire, inserire i relativi parametri di input necessari e impostare una data in cui il package scelto verrà eseguito
- **Rilevanza:** Rilevante

- **Requisito #5**

- **Nome:** Visualizzazione stato dei package schedulati
- **Descrizione:** L'utente deve poter visualizzare lo stato dei package schedulati (a livello di environment)
- **Rilevanza:** Rilevante

3.2.2 Requisiti non funzionali

In questa sezione sono raccolti, seguendo lo schema precedente, i requisiti del sistema cosiddetti non funzionali, ovvero le specifiche di funzionamento che non influiscono direttamente nelle modalità di interazione con il sistema da parte di utenti o entità esterne.

- **Requisito**

- **Nome:** Autenticazione

- **Descrizione:** per la parte di back-end dell'autenticazione viene utilizzata l'autenticazione nativa di SAP BO o quella di windows integrated authentication
- **Rilevanza:** Essenziale

- **Requisito**

- **Nome:** Numero massimo di utenti collegati al sistema
- **Descrizione:** Il numero massimo di utenti che si possono collegare contemporaneamente al sistema è dato dalle capacità del server in cui la parte server side dell'applicazione risiede
- **Rilevanza:** Rilevante

3.2.3 Casi d'uso

Gli utenti che possono accedere al sistema sono solo quelli registrati. Non si da alcuna possibilità di poter creare un nuovo utente!

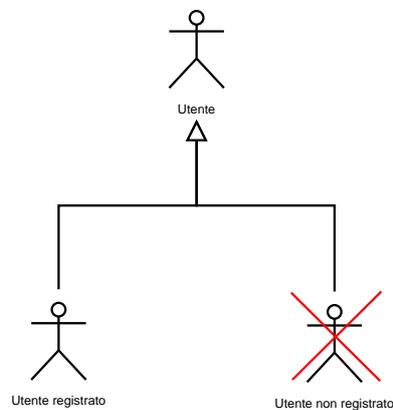


Figura 3.4: Diagramma utenti

Autenticazione

- **Caso d'uso:** Autenticazione utente

- **Descrizione:** L'utente per poter accedere alle funzionalità dell'applicazione si autentica al sistema
- **Attori:** UtenteRegistrato
- **Pre-condizioni:** -
- **Attivazione:** -
- **Scenario:**
 1. il sistema presenta una form con tre campi: server, username e password;
 2. l'utente sceglie il server al quale si vuole connettere;
 3. l'utente inserisce username e password;
 4. l'utente clicca sul pulsante login;
 5. il sistema verifica la correttezza dei dati inseriti;
 6. il sistema autentica l'utente.
- **Alternative:** -
- **Eccezioni:**
 6. username e password inseriti non sono corretti. Torna a 1 con un messaggio d'errore
- **Post-condizioni:** -

Run DTS

- **Caso d'uso:** esecuzione di un package
- **Descrizione:** L'utente esegue un package con gli opportuni parametri in input
- **Attori:** UtenteRegistrato

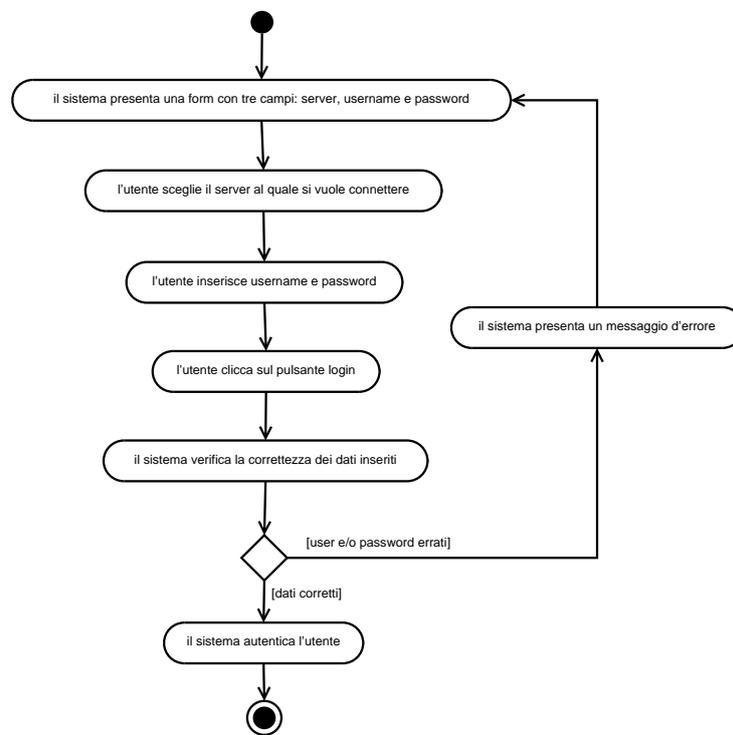


Figura 3.5: Diagramma di attività: autenticazione

- **Pre-condizioni:** l'utente deve aver scelto a quale environment vuole accedere e con quale model vuole lavorare
- **Attivazione:** -
- **Scenario:**
 1. l'utente sceglie, dal menù, run DTS;
 2. l'utente sceglie il package che vuole eseguire;
 3. l'utente inserisce i parametri in input necessari per l'esecuzione del package scelto;
 4. l'utente clicca sul pulsante esecuzione.
- **Alternative:** -

- **Eccezioni:** -
- **Post-condizioni:** -

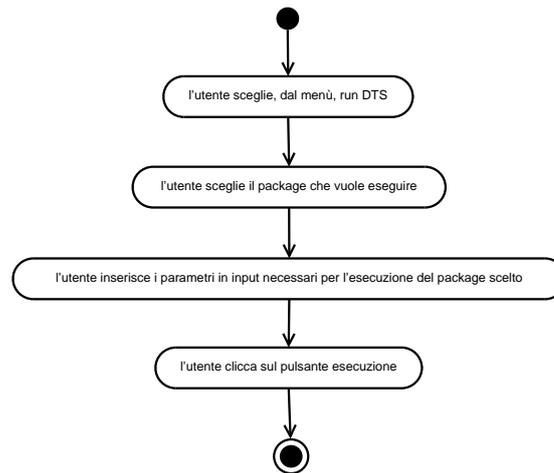


Figura 3.6: Diagramma di attività: esecuzione di un package

View Status

- **Caso d'uso:** visualizzazione stato dei package
- **Descrizione:** L'utente controlla lo stato dei package che sono stati eseguiti
- **Attori:** UtenteRegistrato
- **Pre-condizioni:** l'utente deve aver scelto a quale environment vuole accedere e con quale model vuole lavorare
- **Attivazione:** -
- **Scenario:**
 1. l'utente sceglie, dal menù, view status;

2. a partire dalla lista di package eseguiti (o in esecuzione), l'utente può decidere di vedere i dettagli di ogni singolo elemento della lista;

- **Alternative:** -
- **Eccezioni:** -
- **Post-condizioni:** -

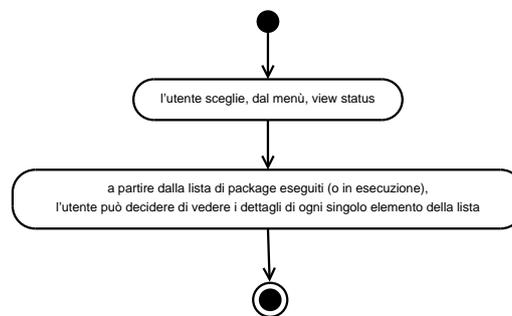


Figura 3.7: Diagramma di attività: visualizzazione dello stato dei package

Schedulazione DTS

- **Caso d'uso:** schedulazione di un package
- **Descrizione:** L'utente schedula un package con gli opportuni parametri in input
- **Attori:** Utente registrato
- **Pre-condizioni:** l'utente deve aver scelto a quale environment vuole accedere e con quale model vuole lavorare
- **Attivazione:** -
- **Scenario:**

1. l'utente sceglie, dal menù, schedulazione DTS;
2. l'utente sceglie il package che vuole schedulare;
3. l'utente inserisce i parametri in input necessari per l'esecuzione del package scelto;
4. l'utente inserisce il giorno e l'ora dell'esecuzione;
5. l'utente sceglie se la schedulazione deve essere unica oppure con una certa frequenza;
6. l'utente clicca sul pulsante schedula.

- **Alternative:** -
- **Eccezioni:** -
- **Post-condizioni:** -

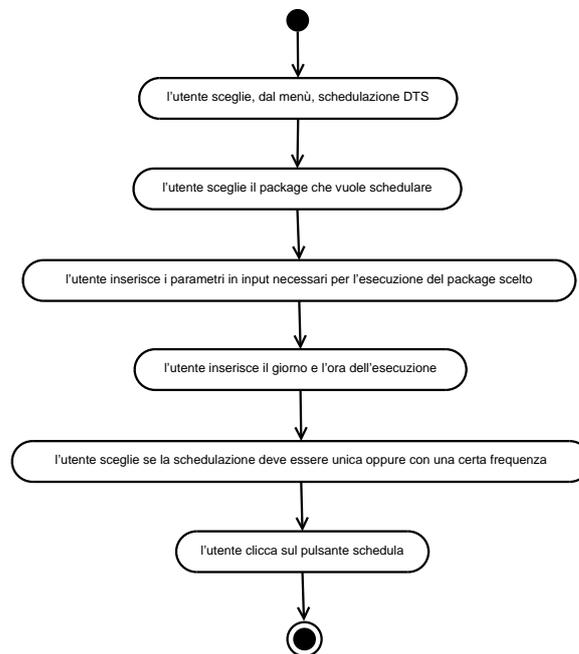


Figura 3.8: Diagramma di attività: schedulazione di un package

Capitolo 4

Architettura del task manager

Le scelte architetturali hanno tenuto conto di due caratteristiche fondamentali che l'applicazione deve avere:

- l'accesso in mobilità dei dispositivi client, permettere cioè ai client di accedere dovunque si trovino;
- la possibilità di utilizzo di diverse tipologie di client, ovvero permettere a diversi dispositivi (possibilmente con diversi sistemi operativi) di accedere all'applicazione.

Viste queste caratteristiche si è scelto di implementare l'applicazione con un'architettura client-server. L'applicazione si compone di due logiche applicative, di business e di presentazione, suddivise su 3 tier distinti, due lato server e uno lato client.

Di seguito viene presentata più in dettaglio l'architettura mostrando il contenuto dei tier.

Architettura 3-tier client-server:

- Tier 0: realizza la logica di presentazione lato client (applicazione iOS);
- Tier 1: realizza la logica di presentazione lato server;
- Tier 2: realizza la vera e propria logica di business del sistema software.

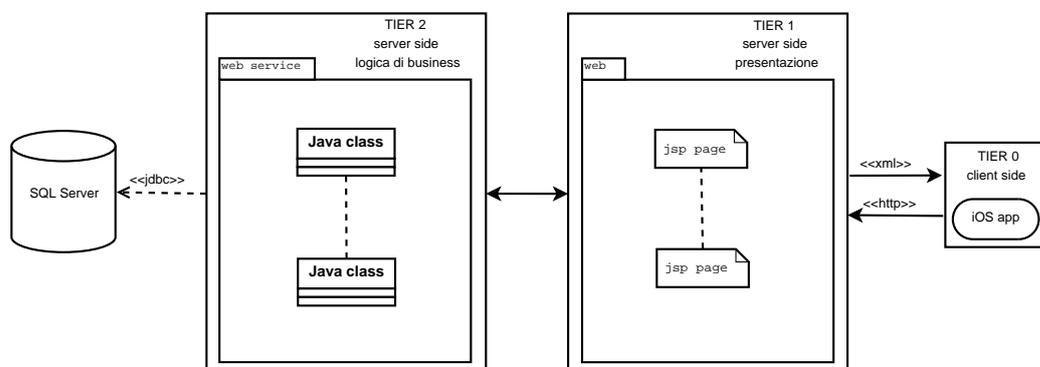


Figura 4.1: Schema architetturale

Vista la natura dell'applicazione, la quale deve permettere l'interoperabilità di diverse tipologie di client (applicazione iOS, applicazione Android, web browser) è stato scelto di implementare il lato server sotto forma di Web Service (tier 2). La comunicazione tra il client e il server avviene attraverso pagine jsp (tier 1). Il client effettua le richieste di tipo https get mentre il server risponde al client attraverso documenti XML.

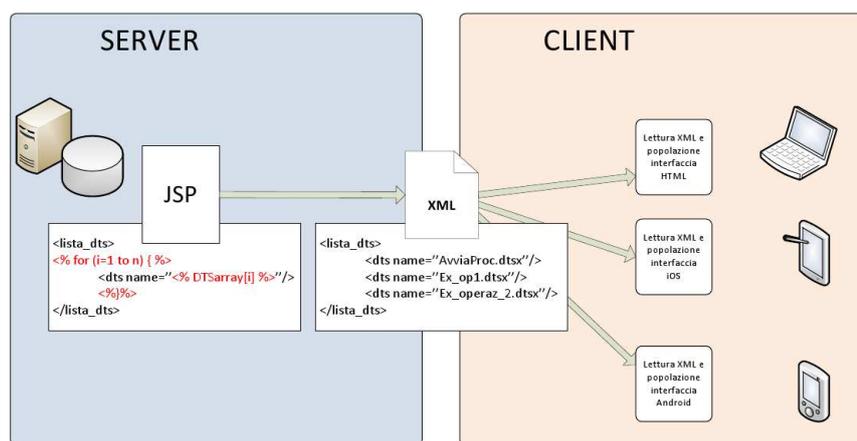


Figura 4.2: Schema architetturale in dettaglio tra tier 0 e tier 1

4.1 JDBC per SQL Server

L'applicazione utilizza un database dal quale recupera tutte le informazioni richieste dal client, il database non è integrato nell'applicazione perché utilizza i dati già presenti in SAP BPC. Questo database può risiedere o sulla stessa macchina su cui gira il Web Service o su una qualunque altra macchina remota.

Visto che ci si è occupati della versione SAP BPC per Microsoft, il database utilizzato è Microsoft SQL Server. Nell'ottica della massima interoperabilità, per interfacciarsi al database, si è deciso di utilizzare le JDBC API quindi anche in caso di cambiamento del database alla base l'applicazione questa non richiederà cambiamenti a livello implementativo.

4.2 Server side (Tier 2)

Nel tier 2 troviamo l'interfaccia del Web Service che permette di eseguire tutte le operazioni previste dalle specifiche. In particolare è possibile:

- aprire e chiudere una connessione al DB;
- controllare l'autenticazione dell'utente;
- ricevere la lista degli environment;
- ricevere la lista dei model;
- ricevere la lista dei DTS;
- ricevere le informazioni sullo stato dei DTS;
- ricevere le informazioni necessarie per l'esecuzione di un DTS;
- eseguire e/o schedulare un DTS.

4.3 Server side (Tier 1)

Nel tier 1 troviamo le pagine jsp che consentono la comunicazione tra il web service e il client, in questo caso l'applicazione iOS.

Come già accennato, il web service e l'applicazione iOS comunicano attraverso documenti XML e tramite richieste https di tipo get. I documenti XML sono fatti in modo che l'applicazione abbia tutte le informazioni che le servono in maniera dinamica.

Di seguito verranno descritte le pagine jsp che compongono l'interfaccia client-server.

index.jsp

Questa è la pagina iniziale dove vengono richiesti in input server, user e password per potersi collegare al sistema di SAP BPC. Essa può restituire tre diversi documenti XML, in base allo stato della variabile di sessione (*new* oppure *old*) oppure perché l'autenticazione non è andata a buon fine.

```
1 <list_next_page>
2   <next_page name="login.jsp">
3   </next_page>
4 </list_next_page>
5
6 <session status="new">
7 </session>
8
9 <list_server variable="server" type="single_choice">
10  <server name="AMSTERDAM" value="AMSTERDAM">
11  </server>
12 </list_server>
13
14 <list_user variable="user" type="free_text">
15 </list_user>
16
17 <list_password variable="password" type="password">
18 </list_password>
```

Listato 4.1: Documento XML: index.jsp (new)

```
1 <list_next_page>
2   <next_page name="menu.jsp">
```

```
3   </next_page>
4 </list_next_page>
5
6 <session status="old">
7   <user name="BPCSystem">
8     </user>
9
10  <server name="AMSTERDAM">
11    </server>
12
13  <environment name="Demo">
14    </environment>
15
16  <model name="Planning">
17    </model>
18
19  <dts name="cDTSCopyActualToOtherCategory.dtsx">
20    </dts>
21 </session>
```

Listato 4.2: Documento XML: index.jsp (old)

```
1 <list_next_page>
2   <next_page name="login.jsp">
3     </next_page>
4 </list_next_page>
5
6 <session status="error" message="Invalid user or password. Please reinsert
   user and password">
7 </session>
8
9 <list_server variable="server" type="single_choice">
10  <server name="AMSTERDAM" value="AMSTERDAM"> </server>
11 </list_server>
12
13 <list_user variable="user" type="free_text">
14 </list_user>
15
16 <list_password variable="password" type="password">
17 </list_password>
```

Listato 4.3: Documento XML: index.jsp (error)

login.jsp

In questa pagina viene effettuato il controllo sull'autenticazione. Essa può restituire due diversi documenti XML, in base alla riuscita o fallimento dell'autenticazione.

```
1 <list_next_page>
2   <next_page name="environment.jsp">
3   </next_page>
4 </list_next_page>
5
6 <session status="authenticated">
7   <user name="EPCSystem">
8   </user>
9
10  <server name="AMSTERDAM">
11  </server>
12 </session>
```

Listato 4.4: Documento XML: login.jsp

```
1 <list_next_page>
2   <next_page name="index.jsp" >
3   </next_page>
4 </list_next_page>
5
6 <session status="error">
7 </session>
```

Listato 4.5: Documento XML: login.jsp (error)

logout.jsp

In questa pagina vengono cancellati tutti i dati della sessione.

```
1 <list_next_page>
2   <next_page name="index.jsp">
3   </next_page>
4 </list_next_page>
```

Listato 4.6: Documento XML: logout.jsp

environment.jsp

In questa pagina viene chiesto all'utente di selezionare un environment tra quelli in cui è autorizzato ad accedere.

```
1 <list_next_page>
2   <next_page name="model.jsp">
3   </next_page>
4 </list_next_page>
5
6 <session>
7   <user name="BPCSystem">
8   </user>
9
10  <server name="AMSTERDAM">
11  </server>
12 </session>
13
14 <list_env variable="environment" type="single_choice" >
15   <env name="Demo" value="Demo">
16   </env>
17
18   <env name="EnvironmentShell" value="EnvironmentShell">
19   </env>
20 </list_env>
```

Listato 4.7: Documento XML: environment.jsp

model.jsp

In questa pagina viene chiesto all'utente di selezionare un model tra quelli in cui è autorizzato ad accedere.

```
1 <list_next_page>
2   <next_page name="menu.jsp">
3   </next_page>
4 </list_next_page>
5
6 <session>
7   <user name="BPCSystem">
8   </user>
9
10  <server name="AMSTERDAM">
11  </server>
12
13  <environment name="Demo">
```

```
14 </environment>
15 </session>
16
17 <list_model variable="model" type="single_choice">
18 <model name="Consolidation" value="Consolidation">
19 </model>
20
21 <model name="Ownership" value="Ownership">
22 </model>
23
24 <model name="Planning" value="Planning">
25 </model>
26
27 <model name="Rates" value="Rates">
28 </model>
29 </list_model>
```

Listato 4.8: Documento XML: model.jsp

menu.jsp

In questa pagina viene chiesto all'utente di scegliere l'operazione che vuole effettuare.

```
1 <list_next_page>
2 <next_page name="dtsList.jsp">
3 </next_page>
4
5 <next_page name="dtsStatus.jsp">
6 </next_page>
7
8 <next_page name="logout.jsp">
9 </next_page>
10 </list_next_page>
11
12 <session>
13 <user name="BPCSystem">
14 </user>
15
16 <server name="AMSTERDAM">
17 </server>
18
19 <environment name="Demo">
20 </environment>
21
22 <model name="Planning">
```

```
23 </model>
24 </session>
```

Listato 4.9: Documento XML: menu.jsp

dtsList.jsp

In questa pagina viene chiesto all'utente di scegliere quale DTS si vuole eseguire.

```
1 <list_next_page>
2   <next_page name="dtsParam.jsp" >
3   </next_page>
4 </list_next_page>
5
6 <session>
7   <user name="BPCSystem">
8   </user>
9
10  <server name="AMSTERDAM">
11  </server>
12
13  <environment name="Demo">
14  </environment>
15
16  <model name="Planning">
17  </model>
18 </session>
19
20 <list_dts variable="DTS" type="single_choice">
21   <dts folder="Data Management"
22     name="FX Restatement"
23     desc="Perform Currency restatement"
24     value="System Files/FX Restatement.dtsx">
25   </dts>
26
27   <dts folder="Data Management"
28     name="Clear"
29     desc="Clear data"
30     value="System Files/Clear.dtsx">
31   </dts>
32
33   ...
34
35   <dts folder="Test"
36     name="cDTSCopyActualToOtherCategory"
```

```
37     desc="DTS test"
38     value="cDTSCopyActualToOtherCategory.dtsx">
39 </dts>
40
41 <dts folder="Data Management"
42     name="Default Formulas"
43     desc="Run default formulas"
44     value="System Files/Default Formulas.dtsx">
45 </dts>
46 </list_dts>
```

Listato 4.10: Documento XML: dtsList.jsp

dtsParam.jsp

In questa pagina viene chiesto all'utente di scegliere i parametri di input per poter eseguire e/o schedulare un DTS.

```
1 <list_next_page>
2   <next_page name="dtsRun.jsp">
3   </next_page>
4
5   <next_page name="dtsSchedule.jsp">
6   </next_page>
7 </list_next_page>
8
9 <session>
10  <user name="BPCSystem">
11  </user>
12
13  <server name="AMSTERDAM">
14  </server>
15
16  <environment name="Demo">
17  </environment>
18
19  <model name="Planning">
20  </model>
21
22  <dts name="cDTSCopyActualToOtherCategory.dtsx">
23  </dts>
24 </session>
25
26 <list_prompt_elem>
27  <prompt_elem message="Select dimension members" info="SELECTINPUT"
    variable="%SELECTION%">
```

```
28 <list_variable variable="Category_set" type="multiple_choice">
29   <elem name="Actual" value="Actual">
30     </elem>
31
32   <elem name="Actual_BudgetRate" value="Actual_BudgetRate">
33     </elem>
34
35   ...
36
37   <elem name="Forecast" value="Forecast">
38     </elem>
39
40   <elem name="Plan" value="Plan">
41     </elem>
42 </list_variable>
43
44 <list_variable variable="Time_set" type="multiple_choice">
45   <elem name="2006.JAN" value="2006.JAN">
46     </elem>
47
48   <elem name="2006.FEB" value="2006.FEB">
49     </elem>
50
51   ...
52   <elem name="2015.OCT" value="2015.NOV">
53     </elem>
54
55   <elem name="2015.NOV" value="2015.DEC">
56     </elem>
57 </list_variable>
58
59 <list_variable variable="Product_set" type="multiple_choice">
60   <elem name="ProductA" value="ProductA">
61     </elem>
62
63   <elem name="ProductB" value="ProductB">
64     </elem>
65
66   <elem name="ProductC" value="ProductC">
67     </elem>
68 </list_variable>
69 </prompt_elem>
70
71 <prompt_elem message="Select whether to run default logic for stored
72   values after importing" info="RADIOBUTTON" variable="%RUNLOGIC%">
73   <list_variable variable="RUNLOGIC" type="single_choice">
74     <elem name="Yes" value="1">
```

```
74     </elem>
75
76     <elem name="No" value="0">
77     </elem>
78 </list_variable>
79 </prompt_elem>
80 </list_prompt_elem>
```

Listato 4.11: Documento XML: dtsParam.jsp

dtsStatus.jsp

In questa pagina viene mostrato, all'utente, lo stato dei DTS eseguiti o in esecuzione.

```
1 <list_next_page>
2   <next_page name="menu.jsp">
3   </next_page>
4 </list_next_page>
5
6 <session>
7   <user name="BPCSystem">
8   </user>
9
10  <server name="AMSTERDAM">
11  </server>
12
13  <environment name="Demo">
14  </environment>
15
16  <model name="Planning">
17  </model>
18 </session>
19
20 <list_dts_status>
21   <dts_status Owner="BPCUser"
22     StartTime="2012-05-24 12:27:54"
23     EndTime="2012-05-24 13:45:14"
24     Operation="Completed"
25     Package="bDTSRunCurrencyConversion"
26     SEQ="9324"
27     Parameters="\%SELECTION%\% ... DIMENSION:TIME|2012.MAR">
28   </dts_status>
29
30   <dts_status Owner="BPCAdmin"
31     StartTime="2012-05-24 11:45:16"
```

```
32         EndTime="2012-05-24 12:24:26"
33         Operation="Completed"
34         Package="bDTS_PxQ_MemoMargins"
35         SEQ="9323"
36         Parameters="\%SELECTION\% ... DIMENSION:Entity|LEB077">
37     </dts_status>
38
39     ...
40
41     <dts_status Owner="BPCSystem"
42         StartTime="2012-04-20 19:50:51"
43         EndTime="2012-04-20 19:54:17"
44         Operation="Completed"
45         Package="bDTSRunDimensionDerivations"
46         SEQ="9310"
47         Parameters="\%SEL\% ... DIMENSION:Entity|LEB077">
48     </dts_status>
49 </list_dts_status>
```

Listato 4.12: Documento XML: dtsStatus.jsp

dtsSchedule.jsp

In questa pagina vengono scelti i parametri necessari alla schedulazione del DTS scelto.

```
1 <list_next_page>
2     <next_page name="dtsRun.jsp" >
3     </next_page>
4 </list_next_page>
5
6 <session>
7     <user name="BPCSystem">
8     </user>
9
10    <server name="AMSTERDAM">
11    </server>
12
13    <environment name="Demo">
14    </environment>
15
16    <model name="Planning">
17    </model>
18
19    <dts name="cDTSCopyActualToOtherCategory.dtsx">
20    </dts>
```

```

21
22   <param name="TIME_SET;2015.OCT,2015.NOV\nCATEGORY_SET;Actual\nPRODUCT_SET;
      ProductA\nRUNLOGIC;1">
23   </param>
24 </session>
25
26 <list_date variable="dateInput" type="date">
27 </list_date>
28
29 <list_hour variable="hourInput" type="hour">
30 </list_hour>
31
32 <list_typeSched variable="typeSched" type="single_choice">
33   <typeSched name="Once" value="Once">
34   </typeSched>
35
36   <typeSched name="Daily" value="Daily">
37   </typeSched>
38
39   <typeSched name="Weekly" value="Weekly">
40   </typeSched>
41
42   <typeSched name="Monthly" value="Monthly">
43   </typeSched>
44 </list_typeSched>

```

Listato 4.13: Documento XML: dtsSchedule.jsp

dtsRun.jsp

In questa pagina viene eseguito il DTS scelto con i parametri in input precedentemente selezionati. Se è presente il tag `<schedule>` allora il DTS invece di essere eseguito subito viene schedulato con i parametri immessi dall'utente.

```

1 <list_next_page>
2   <next_page name="menu.jsp" >
3   </next_page>
4 </list_next_page>
5
6 <session>
7   <user name="BPCSystem">
8   </user>
9
10  <server name="AMSTERDAM">

```

```
11 </server>
12
13 <environment name="Demo">
14 </environment>
15
16 <model name="Planning">
17 </model>
18
19 <dts name="cDTSCopyActualToOtherCategory.dtsx">
20 </dts>
21
22 <param name="TIME_SET;2015.OCT,2015.NOV\nCATEGORY_SET;Actual\nPRODUCT_SET;
    ProductA\nRUNLOGIC;1">
23 </param>
24
25 <schedule name="dateInput;22-05-2012\nhourInput;15:10\ntypeSched;Daily">
26 </schedule>
27 </session>
28
29 <statusRunDts name="ok">
30 </statusRunDts>
```

Listato 4.14: Documento XML: dtsRun.jsp

4.4 Client side (Tier 0)

La parte dell'interfaccia utente è interamente delegata al client il quale la costruisce in base ai documenti XML, descritti nel paragrafo precedente, che riceve dal Web Service.

Anche le risposte alle invocazioni di operazioni vengono restituite dal Web Service sotto forma di documenti XML che vengono parsate dal client, elaborate (se necessario) e mostrate all'utente.

Il client, ricevuto l'input dall'utente, genera un url https, che servirà a invocare il web service con i giusti parametri, i quali vengono ricavati dal documento XML.

I tag più usati nella comunicazione tra il web service e l'applicazione iOS sono:

- `<session>`: all'interno di questo tag vengono elencate tutte le variabili presenti nella session e il relativo valore, al momento in cui viene fatta la richiesta di un documento XML;
- `<next_page>`: viene comunicata la pagina, o le pagine, in cui è possibile dirigersi a partire da quella visualizzata.

Nei tag che servono per la richiesta di dati all'utente ci sono degli attributi che si possono definire "principali" che servono all'applicazione iOS per poter creare l'url con i giusti parametri:

- `variable`: nome della variabile che viene riconosciuta nel metodo get;
- `type`: tipo del campo che deve essere visualizzato all'utente:
 - `free_text`: campo libero, l'utente può scrivere qualunque cosa;
 - `password`: campo offuscato per l'inserimento della password;
 - `single_choice`: da un elenco, l'utente può scegliere un solo elemento, per esempio un radio button;
 - `multiple_choice`: da un elenco, l'utente può scegliere diversi elementi, per esempio una check box;
- `name`: nome dell'elemento, stringa da visualizzare all'utente;
- `value`: valore corrispondente all'attributo `name` che deve essere ritornato con la variabile definita in `variable`

Capitolo 5

Sviluppo applicazione server side

In questo capitolo verrà descritta la parte di sviluppo del lato server di iTwin BPC attraverso la descrizione dell'interfaccia del web service e la spiegazione del funzionamento dell'interprete dei comandi di SAP BPC. Infine verranno presentati tre diagrammi di sequenza riguardanti le tre principali funzioni dell'applicazione: esecuzione, visualizzazione e schedulazione dei DTS.

5.1 Web Service

Il web service è la componente dell'applicazione che realizza la logica di business ed esporta ai client mobili le funzionalità dell'applicazione. Inoltre il web service, attraverso documenti XML, comunica al client tutte le informazioni necessarie sulla struttura che deve avere l'interfaccia utente (tipo di form, numero di radio button, liste di scelta, ecc. . .).

5.1.1 Diagramma delle classi

In questa sezione viene presentato il diagramma che mostra le classi con relativi attributi e metodi che compongono la parte server dell'applicazione.

Vengono inoltre mostrate le dipendenze tra le classi.

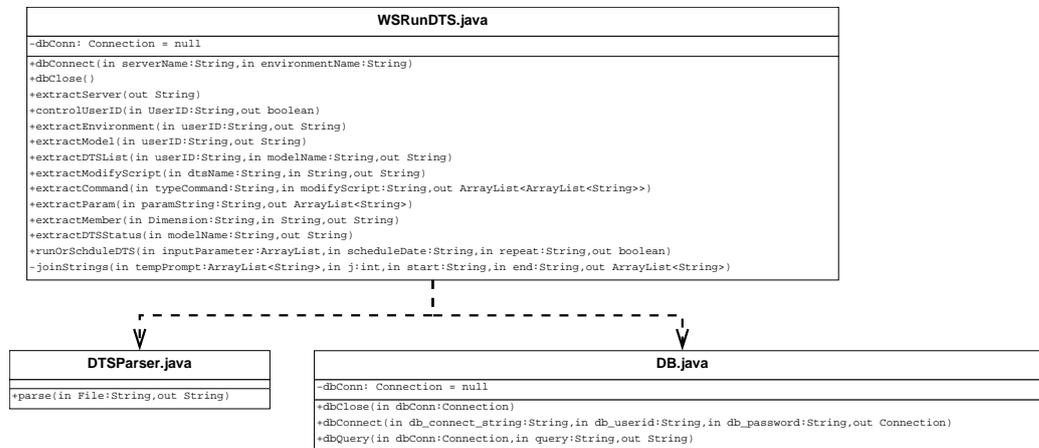


Figura 5.1: Diagramma delle classi del web service

Interfaccia database-web service

L'interfaccia tra il database e il web service viene implementata dalla classe **DB** dove troviamo i metodi pubblici **dbConnect** e **dbClose** che permettono rispettivamente di aprire e chiudere una connessione verso un particolare database e il metodo pubblico **dbQuery** che permette di eseguire una query sul DB a cui ci si è precedentemente connessi. Questo metodo ritorna un errore se la connessione non è stata precedentemente inizializzata. L'attributo privato **dbConn** permette di condividere la connessione al DB tra i metodi della classe.

Interfaccia pubblica del web service

L'interfaccia pubblica del web service viene implementata dalla classe **WSRunDTS** dove vengono implementati la maggior parte dei metodi che servono per l'esecuzione dell'applicazione. Si può dire che questa classe è il "cuore" di iTwin BPC.

Di seguito vengono riportati i metodi che la compongono con una loro breve descrizione:

- **dbConnect** e **dbClose**: anche in questa classe troviamo questi metodi ma a differenza di quelli all'interno della classe **DB**, questi servono solo da tramite per richiamare, scegliendo server e database, i metodi omonimi della classe **DB**;
- **controlUserID**: questo metodo viene utilizzato per controllare i parametri *userID* e *password* utilizzando l'autenticazione di SAP BPC o l'autenticazione di Windows;
- **extractEnvironment** e **extractModel**: questi due metodi effettuano una query nel database per estrarre rispettivamente la lista degli environment e dei model all'interno dell'environment precedentemente selezionato su cui l'utente è abilitato a "lavorare";
- **extractDTSList**: questo metodo effettua una query nel database per estrarre la lista dei DTS relativi al model precedentemente selezionato. Verranno estratti solo i DTS che l'utente è abilitato ad eseguire;
- **extractModifyScript**: questo metodo apre il file corrispondente al DTS precedentemente selezionato. Dopo l'apertura effettua il parsing del file XML utilizzando la classe **DTSParser** per recuperare la parte riguardante la variabile `modifyScript`¹;
- **extractCommand**: questo metodo ricerca all'interno della stringa rappresentante la `modifyScript` l'insieme dei comandi che corrispondono alla stringa passata in input;
- **extractParam**: questo metodo elabora la stringa che rappresenta un parametro dei comandi rendendola in un formato facilmente utilizzabile nella pagina jsp;

¹**modifyScript**: variabile che viene utilizzata da SAP BPC per richiedere i parametri di input all'utente in fase di esecuzione di un DTS

- **extractMember**: questo metodo effettua una query nel database per estrarre i membri di una dimensione;
- **extractDTSStatus**: questo metodo effettua una query nel database per estrarre tutte le informazioni riguardanti i DTS eseguiti e/o in esecuzione nel model precedentemente selezionato dall'utente;
- **runOrScheduleDTS**: questo metodo esegue o schedula il DTS. Se i parametri riguardanti la schedulazione (*scheduleDate* e *repeat*) sono NULL il DTS verrà eseguito, altrimenti verrà schedulato per la data indicata in *scheduleDate* e con una frequenza di esecuzione indicata in *repeat*. Il metodo ritorna TRUE se il DTS va in esecuzione o la schedulazione va a buon fine, FALSE altrimenti;
- **joinString**: questo è l'unico metodo privato della classe. Viene utilizzato dal metodo *extractParam* per una corretta manipolazione delle stringhe estratte dalla modifyScript.

Interprete dei comandi

L'interprete dei comandi di iTWin BPC viene implementato dalla classe **DTSParser**. Questa classe ha al suo interno un solo metodo **parse** che effettua il parser del file XML che rappresenta il DTS da lanciare ed estrae la parte riguardante la ModifyScript.

Questo interprete replica le funzionalità dell'interprete di SAP BPC, ovvero dapprima analizza la variabile ModifyScript contenuta nei DTS, dalla quale estrae le informazioni necessarie all'esecuzione del DTS (parametri di input, impostazioni delle varie connessioni, ecc. . .) e successivamente lo esegue con gli opportuni parametri passati dall'utente (attraverso il client).

Di seguito vengono descritti i principali comandi della ModifyScript parsi dall'interprete di iTWin BPC raggruppati in base alla loro tipologia.

PROMPT

Questa categoria di comandi serve per richiedere all'utente delle informazioni necessarie per l'esecuzione del DTS.

```
PROMPT(INFILES, [variable], [label])
```

```
PROMPT(OUTFILE, [variable], [label])
```

Questi due comandi vengono utilizzati per richiedere all'utente qual è il file, rispettivamente di input e output, da utilizzare all'interno del DTS.

```
PROMPT(SELECT, [variable], [second variable], [label], [dimensions])
```

```
PROMPT(SELECTINPUT, [variable], [second variable], [label], [dimensions])
```

Questi due comandi servono per richiedere all'utente un membro, o un insieme di membri, di una determinata dimensione. Con lo stesso comando possono essere richiesti all'utente i membri di più dimensioni.

```
PROMPT(CHECKBOX, [variable], [label], [default value])
```

```
PROMPT(CHECKBOXYES, [variable], [label])
```

```
PROMPT(CHECKBOXNO, [variable], [label])
```

```
PROMPT(CHECKBOXGROUP, [variable], [label], [default value], [label List], [value List])
```

Questi quattro comandi consentono di mostrare all'utente una lista di elementi utilizzando una lista di checkbox permettendogli così di selezionare più di un elemento.

```
PROMPT(RADIOBUTTON, [variable], [label], [default value], [label List], [value List])
```

Questo comando consente di mostrare all'utente una lista di elementi utilizzando una lista di radiobutton permettendogli così di selezionare esclusivamente un elemento tra quelli mostrati.

OTHER

In questa categoria troviamo tutti i comandi che non rientrano nella precedente categoria.

```
TASK([Task Name], [PropertyName], [PropertyValue])
```

Questo comando consente di modificare dinamicamente le proprietà di un task del DTS.

```
CONNECTION([ConnectionName], [OleDbPropertyName], [PropertyValue])
```

```
OLEDBCONN([ConnectionName],[OleDbPropertyName], [PropertyValue])
```

Questi comandi consentono di modificare dinamicamente le proprietà di una connessione del DTS.

```
Global([variablename], [value])
```

Questo comando consente di assegnare un valore ad una variabile del DTS.

5.2 Presentazione web

La parte riguardante la presentazione web la troviamo nel tier 1. Questa parte è stata sviluppata attraverso pagine jsp che permettono la comunicazione tra il web service e il client. Le pagine jsp contengono le chiamate ai metodi del web service. Ottenute le risposte alle chiamate dei metodi viene creato un documento XML “ad hoc” che servirà al client per generare l’interfaccia utente.

La struttura dei documenti XML che vengono creati dalle pagine jsp è stata mostrata nel paragrafo 4.3.

5.3 Diagrammi di sequenza

In questa sezione verranno presentati alcuni diagrammi di sequenza relativi alle funzioni utilizzate dagli utenti.

Fino al menù per la scelta dell’operazione da effettuare la sequenza di operazioni è la stessa: si inizia con la scelta del server a cui ci si vuole collegare e il relativo login. Successivamente verrà richiesto qual è l’environment di

riferimento e a seguire il model da considerare. Dopo queste attività verrà presentato un menù di scelta in cui si potrà decidere di eseguire un DTS oppure guardare lo stato dei DTS già eseguiti e/o in esecuzione nel relativo model. Scegliendo l'esecuzione del DTS, prima di completare l'operazione, verrà mostrato un altro menù nel quale si chiederà all'utente di scegliere se vuole eseguire subito il DTS oppure schedarlo.

5.3.1 Esecuzione di un DTS

Nel caso in esame l'utente avrà scelto di eseguire un DTS: gli verrà proposta una lista di possibili DTS e l'utente dovrà scegliere quale eseguire. Scelto il DTS gli verranno richiesti i parametri necessari per il lancio e dopo un breve riepilogo dei dati fin'ora inseriti, gli verrà mostrato un piccolo menù nel quale l'utente potrà scegliere di eseguire il DTS e questo verrà mandato subito in esecuzione.

5.3.2 Schedulazione di un DTS

In questo caso l'utente nel secondo menù sceglierà di schedulare il DTS. Gli verrà richiesto di inserire una data e un orario nel quale il DTS dovrà essere eseguito e gli viene altresì chiesto se il lancio dovrà essere ripetuto con una certa frequenza.

5.3.3 View status dei DTS

In quest'ultimo caso, dopo il primo menù di scelta, l'utente sarà in grado di visualizzare i DTS eseguiti e/o in esecuzione relativi al model precedentemente selezionato.

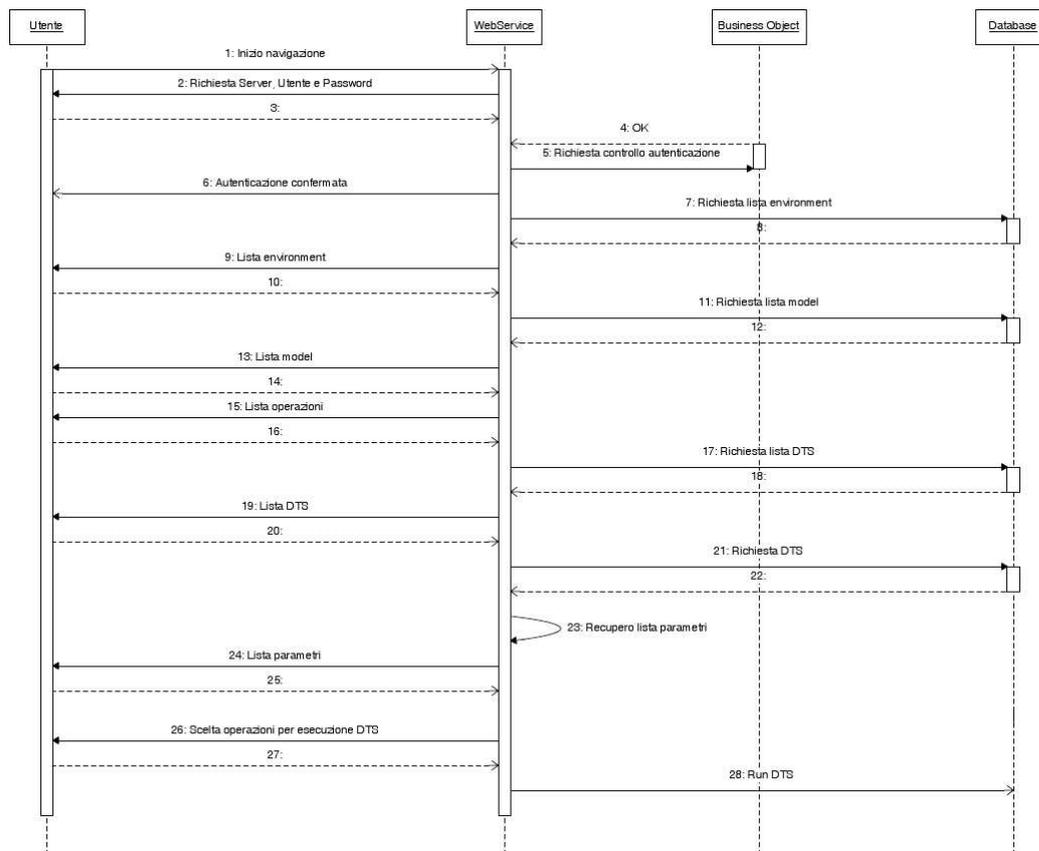


Figura 5.2: Diagramma di sequenza: esecuzione di un DTS

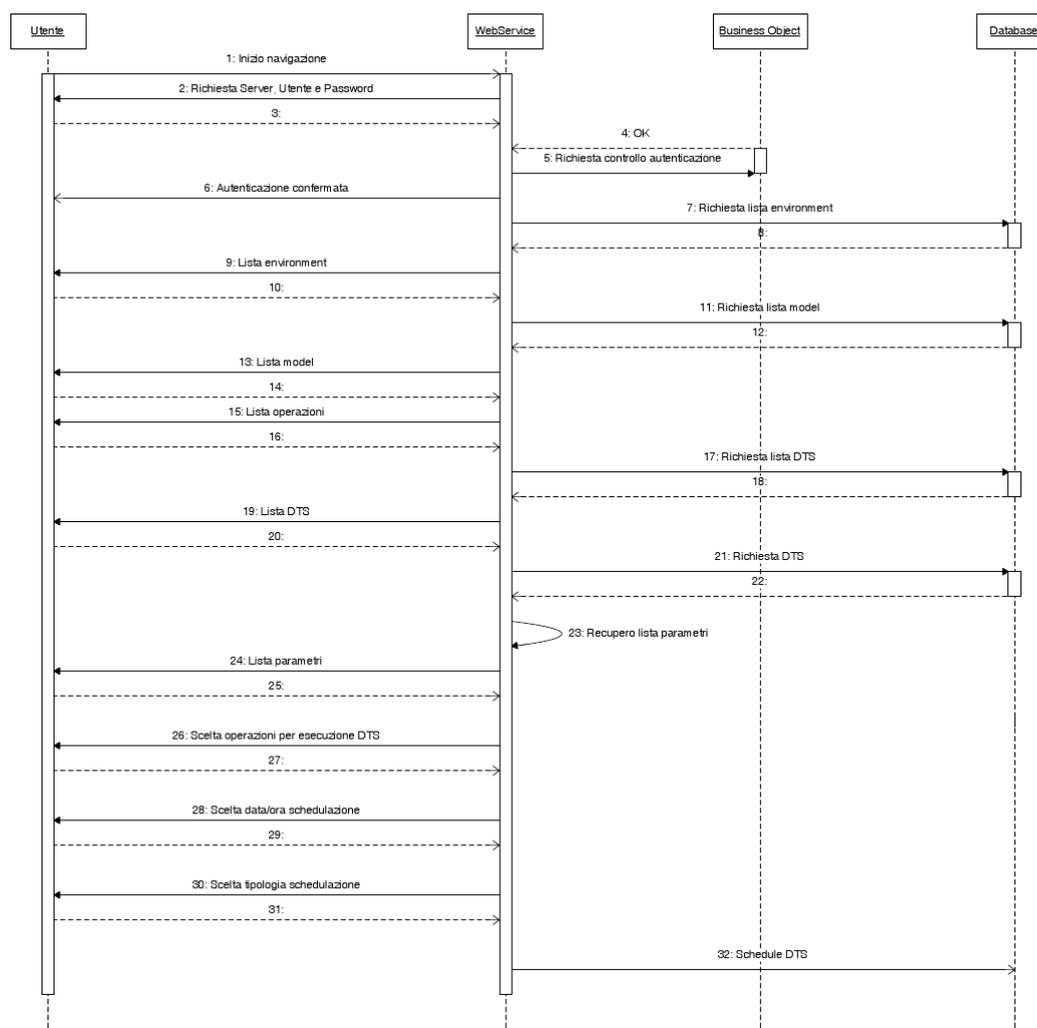


Figura 5.3: Diagramma di sequenza: schedulazione di un DTS

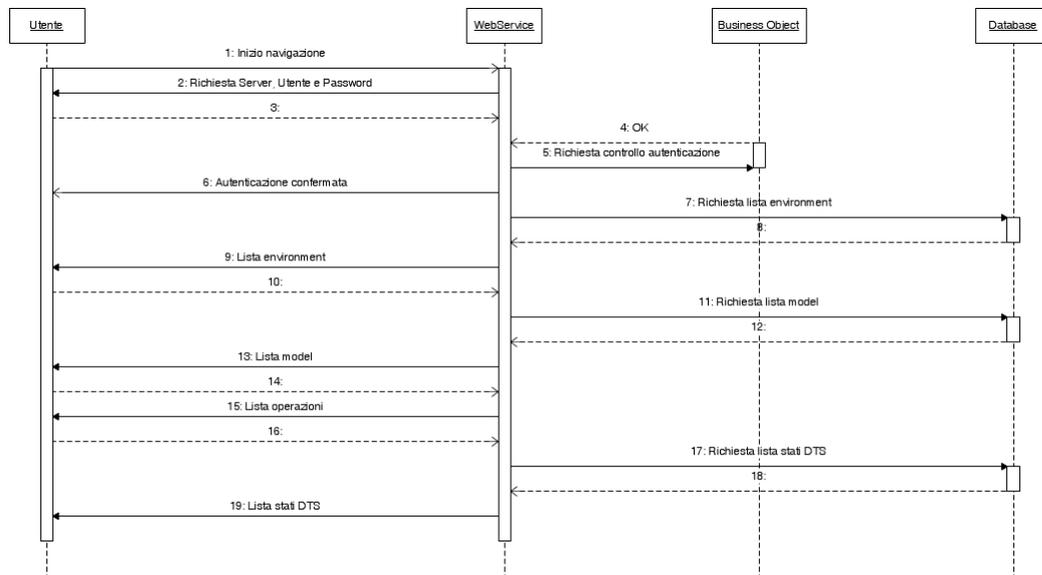


Figura 5.4: Diagramma di sequenza: view status dei DTS

Capitolo 6

Sviluppo interfaccia client-side

La parte client di iTwin BPC è un'applicazione iOS che è stata sviluppata separatamente in collaborazione con un gruppo di lavoro di Twinergy dedicato allo sviluppo di applicazioni iOS. Per il momento questa applicazione è l'unico client per iTwin BPC ma grazie alla strategia multi-piattaforma utilizzata per la parte di interfaccia dell'applicazione sarà possibile avere client per ogni piattaforma (Android, Windows mobile, ecc. . .).

Lo sviluppo è stato suddiviso in 2 fasi:

1. **demo off-line**: la prima fase è stata dedicata alla costruzione di una versione di demo che simulasse il funzionamento dell'applicazione completa. Venivano presentate tutte le schermate di selezione dell'utente: login, scelta environment, scelta model, scelta DTS, selezione parametri di esecuzione, simulazione di lancio DTS e consultazione degli stati dei DTS. Questa prima versione non aveva la possibilità di collegarsi al web service: il funzionamento e le scelte effettuabili dall'utente venivano simulate attraverso un set di documenti xml creati appositamente per la demo e salvati in locale sul dispositivo. In questo modo è stato possibile fornire un'esperienza d'uso reale da mostrare internamente all'azienda e ai potenziali clienti;

2. **prima release funzionante:** in questa fase si è proseguito con lo sviluppo dell'applicazione con l'obiettivo di arrivare ad una prima release completamente funzionante. Si sta sviluppando la parte di connessione al web service e l'arricchimento delle funzionalità dell'applicazione come per esempio il salvataggio delle connessioni preferite, la scelta dei parametri di schedulazione dei DTS (data, periodicità, ecc...), l'ordinamento e la navigazione schermata dello stato dei DTS.

6.1 Paradigma di sviluppo

Pur mantenendo lo sviluppo object oriented, si è adottato il paradigma Model-View-Controller (MVC) come consigliato dalle linee guida di Apple. In questo modo si ha separazione tra il modello logico dell'applicazione e le viste dell'interfaccia grafica permettendo così di avere una vista specifica per ogni tipo di dispositivo (iPhone e iPad) senza andare a modificare la logica dell'applicazione.

Per questioni di economia l'interfaccia grafica attualmente in sviluppo è per iPad ma è stata pensata per non essere troppo diversa da quella che sarà per iPhone. Quando si realizzerà la versione per iPhone basterà duplicare le Viste e i Controller attuali e adattarli al nuovo tipo di dispositivo. Non sarà necessario modificare nulla nel codice del Model.

6.2 Tecnologie utilizzate

L'applicazione è stata realizzata utilizzando l'SDK di Apple per iOS (versione 5.0 o superiore). Il linguaggio di programmazione è Objective-C mentre l'ambiente di sviluppo XCode.

Per ottimizzare l'uso della memoria è stata utilizzata la tecnologia ARC¹. Il sistema operativo mobile di Apple non dispone di meccanismi di garbage collection per la rimozione degli oggetti dalla memoria quando questa è pie-

¹ARC: Automatic Reference Counting

na. Con ARC viene mantenuto in memoria, per ogni oggetto, il numero dei riferimenti che vi puntano; non appena un oggetto non ha nessun riferimento viene rimosso dalla memoria.

Per la progettazione dell'interfaccia grafica è stato utilizzato Storyboards². In questo modo la gestione della navigazione tra le viste non deve essere svolta via codice ma in maniera molto più intuitiva.

6.3 Diagramma delle classi

Le classi utilizzate nello sviluppo dell'applicazione sono di due tipi:

- **Model:** questo tipo di classe contiene variabili e metodi legati al funzionamento dell'applicazione
- **ViewControllers:** questo tipo di classe contiene i controller delle viste che comunicano con il model.

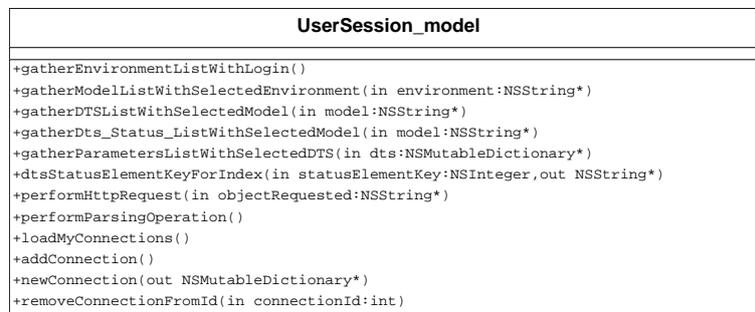


Figura 6.1: Diagramma delle classi dell'applicazione iOS (Model)

La chiamata HTTP, il parsing dell'XML e la gestione delle variabili vengono effettuati dal model.

Il popolamento dell'interfaccia è gestito dal view controller della vista in questione. Ad esempio la vista di scelta del model è Model_3_ViewController.h/.m.

²**Storyboards:** strumento che consente la definizione grafica delle varie viste dell'applicazione, del loro legame e delle transizioni tra esse.

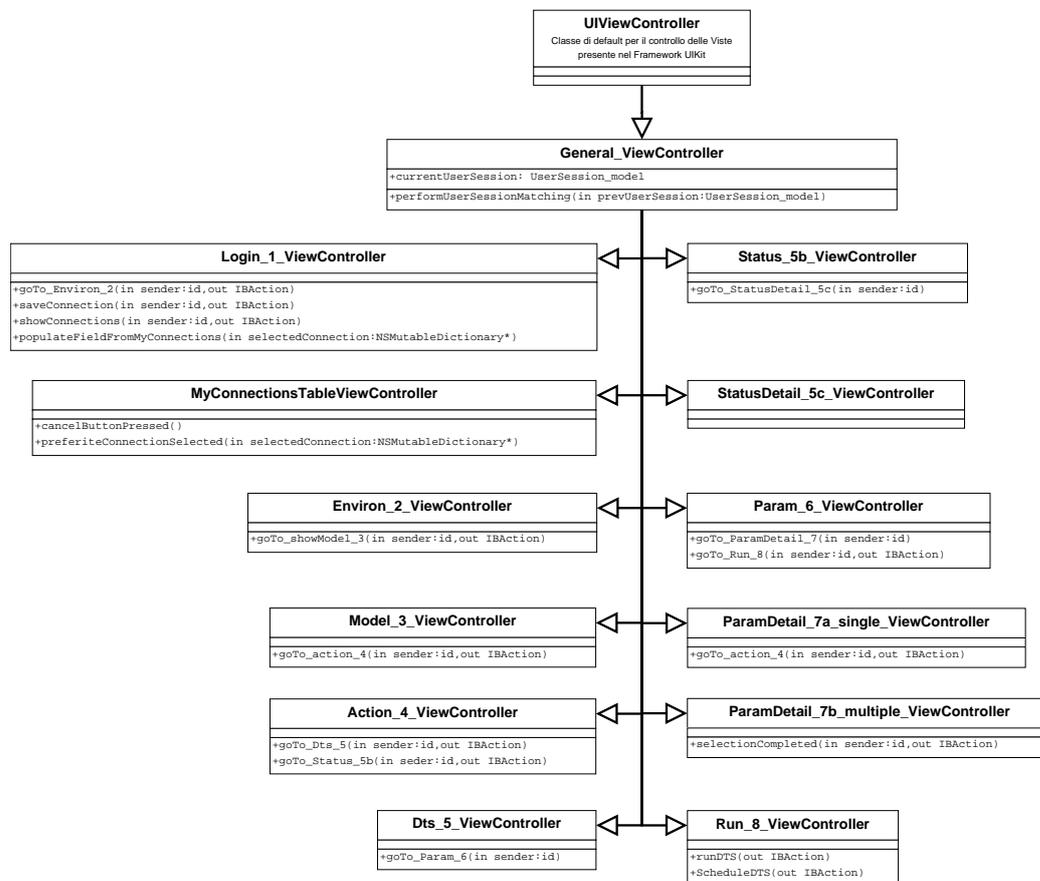


Figura 6.2: Diagramma delle classi dell'applicazione iOS (ViewControllers)

Il numero 3 indica che è la terza schermata vista dall'utente. La classe di ogni view controller ha un riferimento al model `UserSession_model`.

6.3.1 Model

In questo tipo di classe troviamo il singleton³ `UserSession_model` che rappresenta lo stato attuale dell'utilizzo dell'applicazione dell'utente. Questa classe contiene variabili di istanza per memorizzare le selezioni attuali di

³**Singleton**: design pattern creazionale che ha lo scopo di garantire che di una determinata classe venga creata una e una sola istanza, e di fornire un punto di accesso globale a tale istanza.

login, environment e model⁴. Se l'utente sta per lanciare un nuovo DTS, la classe `UserSession_model` memorizza quale DTS è stato selezionato e i valori di input necessari per la sua esecuzione.

La classe `UserSession_model` riceve messaggi dalle classi di controllo delle viste (ViewControllers) per la scrittura o la lettura di parametri (per esempio l'environment correntemente selezionato) o per richiedere l'esecuzione di metodi (per esempio per la connessione https o il parsing).

I metodi che troviamo in questa classe sono:

- **gatherEnvironmentListWithLogin**: questo metodo esegue una chiamata al Web Service per reperire la lista degli environment;
- **gatherModelListWithSelectedEnvironment**: questo metodo esegue una chiamata al Web Service per reperire la lista dei model;
- **gatherDTSListWithSelectedModel**: questo metodo esegue una chiamata al Web Service per reperire la lista dei DTS;
- **gatherDts_Status_ListWithSelectedModel**: questo metodo esegue una chiamata al Web Service per reperire la lista degli status dei DTS;
- **gatherParametersListwithSelectedDTS**: questo metodo esegue una chiamata al Web Service per reperire la lista dei parametri da settare per il DTS corrente;
- **dtsStatusElementKeyForIndex**: questo metodo viene utilizzato per mostrare lato utente la lista dei parametri in un ordine prefissato. Dato un indice intero, restituisce il nome della proprietà del DTS in quella posizione;
- **performHttpRequest**: questo metodo smista le chiamate al web service: in input si aspetta una stringa che rappresenta cosa richiedere

⁴In questo caso con model si intende il cubo di analisi all'interno dell'environment

e successivamente lancia una connessione http appropriata e setta la variabile di istanza *currentOBJRequest*;

- **performParsingOperation**: questo metodo smista le operazioni di parsing. Controlla il valore della variabile *currentObjectRequest* e lancia l'esatto metodo di parsing;
- **addConnection**: questo metodo aggiunge una connessione al file *.plist*⁵;
- **loadMyConnection**: questo metodo esegue un refresh delle connessioni salvate in maniera permanente, leggendole dal file *.plist*;
- **removeConnectionFromId**: questo metodo rimuove una connessione dal file *.plist*.

6.3.2 ViewControllers

Come abbiamo già accennato, in questo tipo di classe troviamo i controller delle viste che comunicano con il Model. Per ogni vista è stata implementata una classe che ne definisce il comportamento tramite la logica “delegate” (gestione delle tabelle, gestione dei UIPickerView, ecc. . .); i metodi in comune tra le viste che consentono la loro comunicazione e la gestione delle immagini, invece, sono stati implementati in una nella classe principale che viene ereditata dalle singole classi rappresentanti le viste. Uno dei metodi in comune è **performUserSessionMatching** che imposta, prima della visualizzazione della vista, il riferimento all'istanza di *UserSession_model* copiandolo dalla vista precedente.

Per ognuna delle schermate, le operazioni effettuate sono:

- chiamata HTTP al web service con gli opportuni parametri;

⁵.**plist**: Property List. Tipo di file fisico del framework Cocoa in iOS e Mac OS X che, in genere, contiene le preferenze dell'utente.

- ricezione dal web service della risposta sotto forma di un documento XML;
- parsing del documento XML;
- popolamento dell'interfaccia grafica utilizzando i dati ricavati dal documento XML.

Qui di seguito verranno descritte le classi e le relative viste che compongono l'interfaccia iOS.

Login_1_ViewController

In questa classe troviamo i seguenti metodi:

- **goTo_Environ_2**: questo metodo permette il passaggio alla vista successiva;
- **saveConnection**: questo metodo legge le credenziali inserite dall'utente e chiama il metodo `addConnection` dell'istanza di `UserSession_model`;
- **showConnections**: questo metodo mostra la schermata di scelta delle connessioni che l'utente ha precedentemente salvato come preferite (`MyConnections`);
- **populateFieldFromMyConnections**: questo metodo inserisce nella caselle di input le credenziali che l'utente ha selezionato dalla lista `MyConnections`.

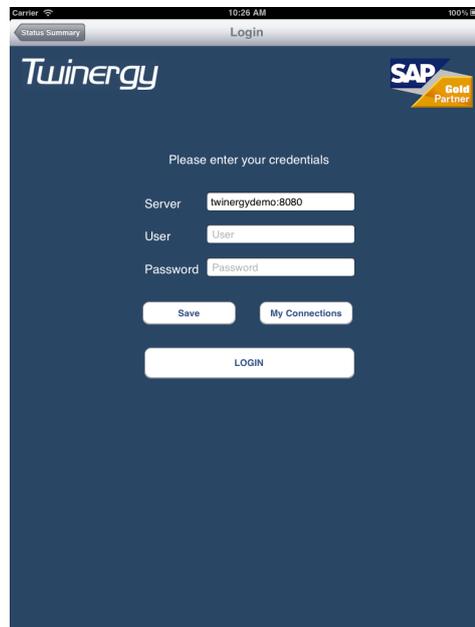


Figura 6.3: Screenshot applicazione: login

MyConnectionsTableViewController

In questa classe troviamo i metodi per gestire la vista della tabella delle MyConnection. Tra questi metodi possiamo metterne in evidenza due: **cancelButtonPressed** e **preferiteConnectionSelected**, che permettono di delegare alla precedente vista, rispettivamente, la “scomparsa” della tabella dallo schermo e l’inserimento dei valori selezionati nelle caselle di testo.

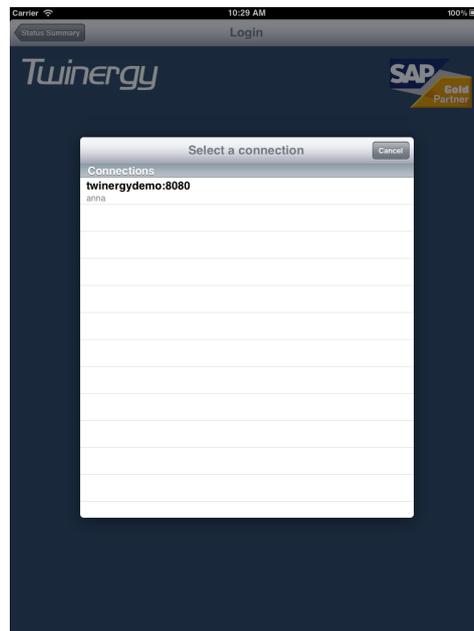


Figura 6.4: Screenshot applicazione: lista connessioni salvate

Environ_2_ViewController

In questa classe troviamo solo il metodo `goTo_showModel_3` che permette il passaggio alla vista successiva.



Figura 6.5: Screenshot applicazione: lista environment

Model_3_ViewController

In questa classe troviamo solo il metodo **goTo_action_4** che permette il passaggio alla vista successiva.

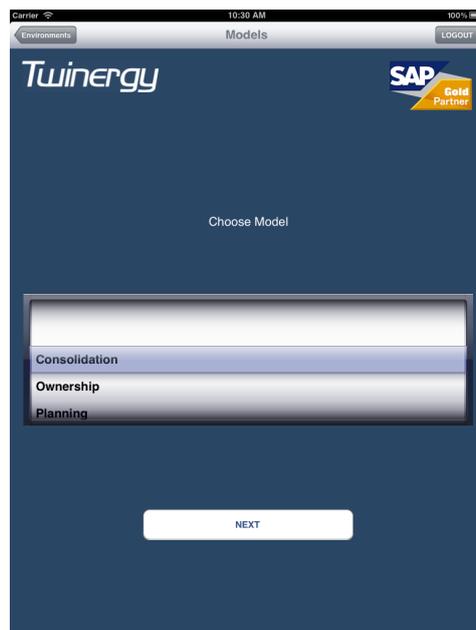


Figura 6.6: Screenshot applicazione: lista model

Action_4_ViewController

In questa classe troviamo i metodi che permettono il passaggio alle viste successive.

- **goTo_Dts_5**: questo metodo permette il passaggio alla vista che consente la scelta del DTS da eseguire;
- **goTo_Status_5b**: questo metodo permette il passaggio alla vista che contiene la lista degli stati dei DTS eseguiti e/o in esecuzione.

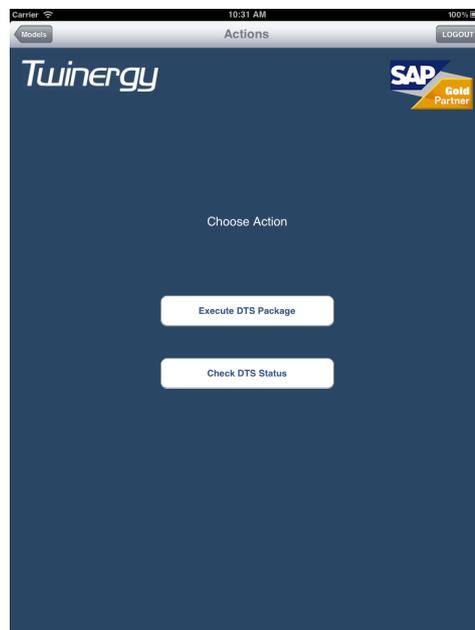


Figura 6.7: Screenshot applicazione: menù

Dts_5_ViewController

In questa classe troviamo solo il metodo `goTo_Param_6` che permette il passaggio alla vista successiva.

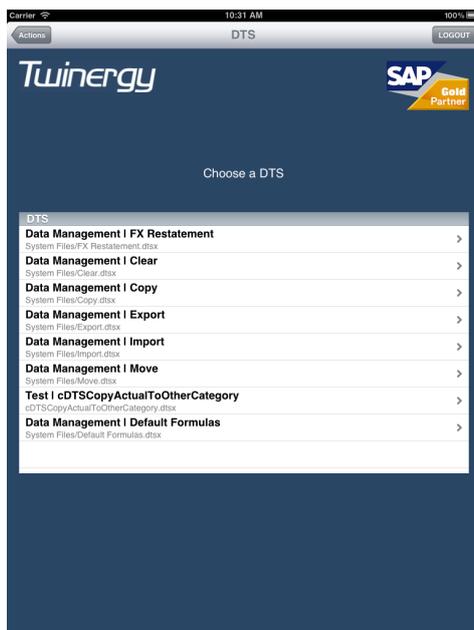


Figura 6.8: Screenshot applicazione: lista DTS

Status_5b_ViewController

In questa classe troviamo solo il metodo `goTo_StatusDetail_5c` che permette il passaggio alla vista successiva.

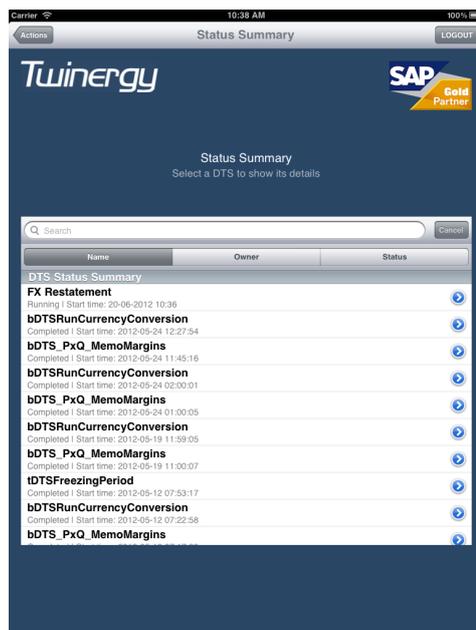


Figura 6.9: Screenshot applicazione: lista stati DTS

Status_5c_ViewController

In questa classe non sono presenti metodi specifici.

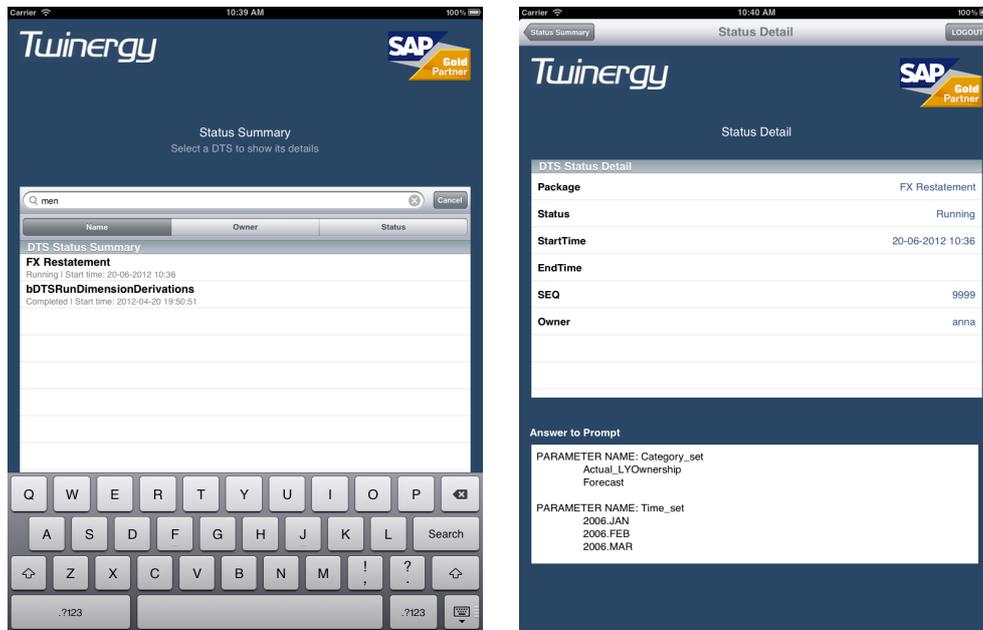


Figura 6.10: Screenshot applicazione: lista stati DTS con ordinamento e in dettaglio

Param_6_ViewController

In questa classe troviamo i metodi che permettono il passaggio alle viste successive.

- **goTo_ParamDetail_7**: questo metodo permette il passaggio alla vista che consente di selezionare le informazioni riguardanti i parametri necessari per l'esecuzione del DTS;
- **goTo_Run_8**: questo metodo permette il passaggio alla vista che contiene il riepilogo delle informazioni immesse per l'esecuzione del DTS.

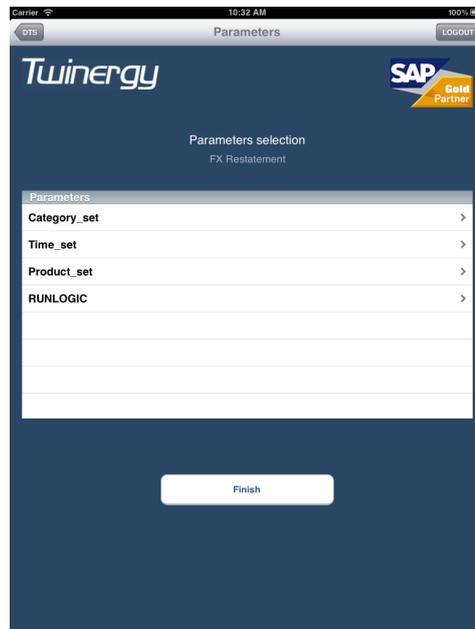


Figura 6.11: Screenshot applicazione: lista parametri DTS

ParamDetail_7a_single_ViewController

In questa classe troviamo solo il metodo **goTo_action_4** che permette il passaggio alla vista successiva.

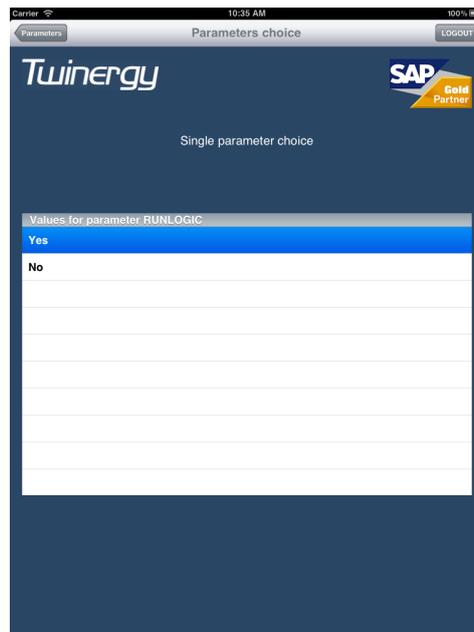


Figura 6.12: Screenshot applicazione: scelta singola parametri

ParamDetail_7b_multiple_ViewController

In questa classe troviamo solo il metodo **selectionCompleted** che permette di catturare l'evento della pressione del tasto che indica la terminazione della selezione di parametri multipli.

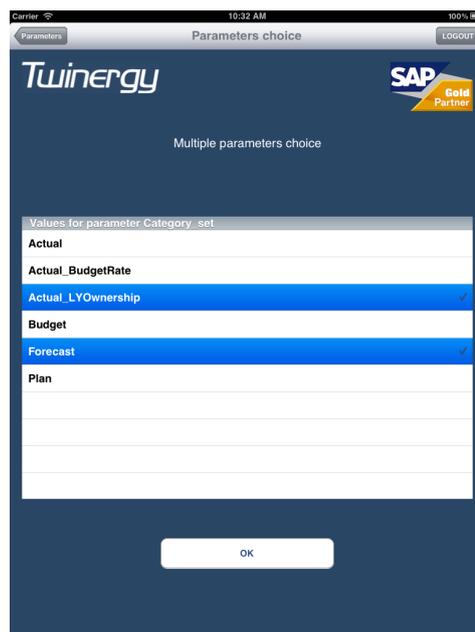


Figura 6.13: Screenshot applicazione: scelta multipla parametri

Run_8_ViewController

In questa classe troviamo i metodi che permettono l'esecuzione o la schedulazione di un DTS.

- **runDTS**: questo metodo permette di catturare la scelta dell'utente di eseguire il DTS;
- **scheduleDTS**: questo metodo permette di catturare la scelta dell'utente di schedulare un DTS.

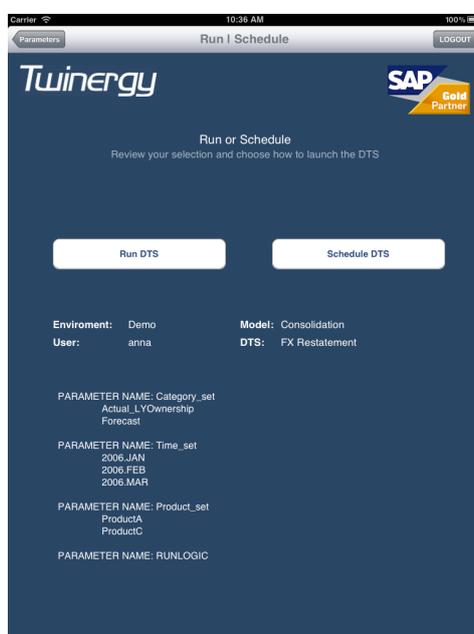


Figura 6.14: Screenshot applicazione: menu run/schedule

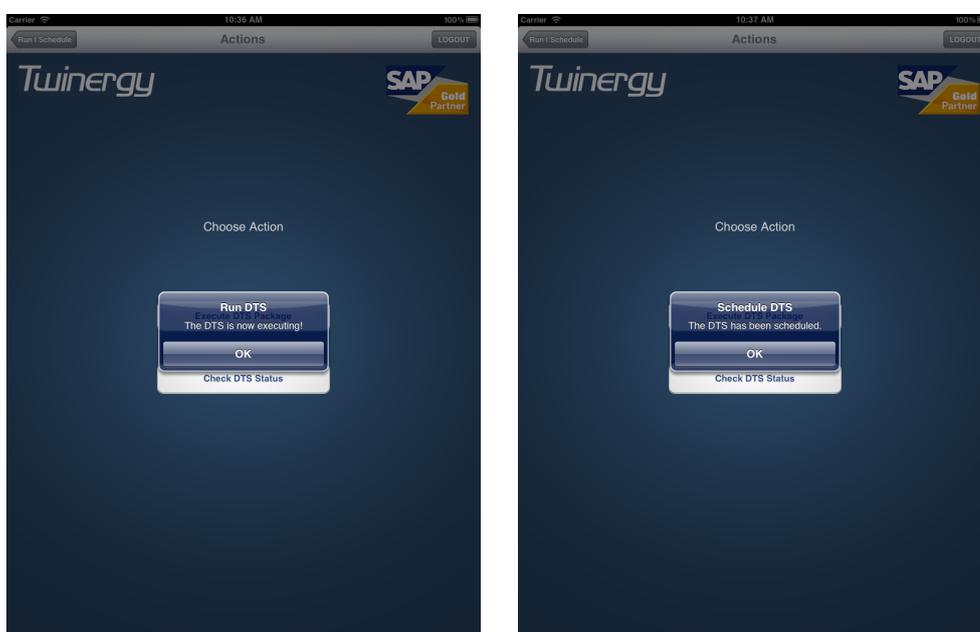


Figura 6.15: Screenshot applicazione: esecuzione e schedulazione DTS

Conclusioni e sviluppi futuri

Per questo lavoro di tesi si è dapprima fatto uno studio sui dispositivi mobili e la loro influenza nell'ambito della BI. Si è continuato esponendo la strategia mobile della SAP per il 2011-2012 per poi passare allo sviluppo di iTwin BPC, un'applicazione mobile che consente l'utilizzo di alcune funzionalità di SAP BPC per Microsoft, in particolare l'esecuzione o la schedulazione di un DTS e la visualizzazione dello stato dei DTS eseguiti o in esecuzione.

Lo sviluppo ha richiesto, in una prima fase, lo studio delle operazioni compiute da SAP BPC per l'esecuzione di un DTS. Questo studio ha portato a sviluppare inizialmente un parser che estraesse dal file XML, che rappresenta il DTS, tutti i parametri necessari all'esecuzione del DTS stesso, questo è il nucleo di iTwin BPC. Successivamente è stato sviluppato il web service e l'applicazione client che permettono da iPhone e iPad, attraverso l'autenticazione al sistema di SAP BPC, di eseguire o schedulare un DTS e di vedere lo stato dei DTS eseguiti e/o in esecuzione.

Attualmente iTwin BPC è in fase di testing, il rilascio di prima release funzionante è stato fissato per luglio 2012.

Di seguito verranno presentati alcuni possibili sviluppi futuri per iTwin BPC.

Attualmente l'applicazione lavora esclusivamente con SAP BPC per Microsoft, si prevede di estenderla con il supporto a NetWeaver⁶.

Per quanto riguarda la parte client di iTwin BPC, attualmente l'unico client è l'applicazione iOS ma si prevede di svilupparne ulteriori, inizialmente per Android e Blackberry e successivamente anche per altre piattaforme.

Infine si prevede di poter estendere le funzionalità dell'applicazione. iTwin BPC verrà presentata con un insieme di funzionalità base che potranno essere integrate su specifica richiesta del cliente, fornendogli così un'applicazione "ad hoc".

A titolo esemplificativo, una delle funzionalità aggiuntive che è stata richiesta durante le presentazioni delle demo, e che quindi verrà molto probabilmente integrata nel pacchetto base, è la possibilità, da parte dell'utente, di gestire i DTS che sono stati schedulati e sono in attesa di esecuzione ovvero visualizzarli ed eventualmente modificarne e/o cancellarne la schedulazione.

⁶**SAP NetWeaver:** piattaforma integrata della SAP che sta alla base di molte applicazioni SAP fin dalla SAP Business Suite. SAP NetWeaver fornisce l'ambiente di sviluppo e di runtime per le applicazioni SAP e può essere utilizzato per l'integrazione con altri sistemi ed applicazioni.

Appendice A

Web Service

Secondo la definizione data dal W3C¹ un Web Service è un sistema software progettato per supportare l'interoperabilità tra macchine connesse in rete.[10] I Web Service sono caratterizzati da:

- un'interfaccia che ne espone il funzionamento descritta mediante uno specifico formato: WSDL²;^[11]
- un protocollo per lo scambio dei messaggi per la comunicazione con altri sistemi: SOAP³;
- formattazione dei messaggi SOAP utilizzando XML;
- scambio dei messaggi SOAP tramite protocollo HTTP;
- pubblicazione del web service in un registry centralizzato chiamato UDDI⁴.

Proprio grazie all'utilizzo di standard basati su XML, tramite un'architettura basata sui Web Service (chiamata, con terminologia inglese, Service Oriented Architecture) applicazioni software scritte in diversi linguaggi di

¹**W3C**: World Wide Web Consortium

²**WSDL**: Web Service Definition Language

³**SOAP**: Simple Object Access Protocol

⁴**UDDI**: Universal Description Discovery and Integration

programmazione e implementate su diverse piattaforme hardware possono quindi essere utilizzate, tramite le interfacce che queste “espongono” pubblicamente e mediante l’utilizzo delle funzioni che sono in grado di effettuare (i “servizi” che mettono a disposizione) per lo scambio di informazioni e l’effettuazione di operazioni complesse (quali, ad esempio, la realizzazione di processi di business che coinvolgono più aree di una medesima azienda) sia su reti aziendali come anche su Internet: la possibilità dell’interoperabilità fra diversi linguaggi di programmazione (ad esempio, tra Java e Python) e diversi sistemi operativi (come Windows e Linux) è resa possibile dall’uso di standard “aperti”.

Di seguito viene mostrato un diagramma che mostra l’interazione tra Client (Service Requester), Web Service (Service Provider) e Registry (Service Broker):

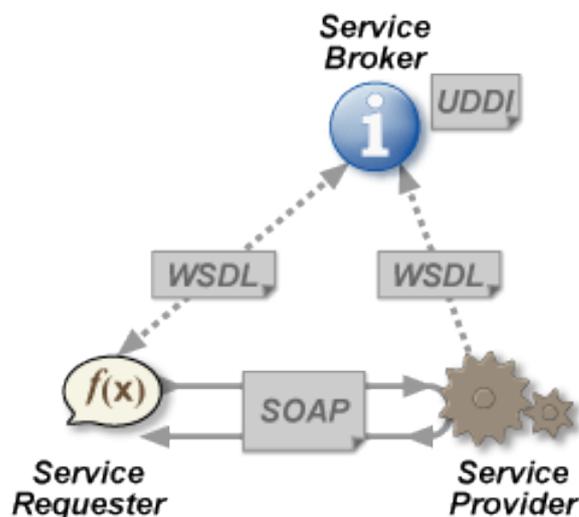


Figura A.1: Interazione tra Client, Web Service e Registry

A.1 WSDL

Il Web Service Description Language è un linguaggio in formato XML utilizzato per descrivere le interfacce dei Web Service, le informazioni contenute in un documento WSDL sono:

- le operazioni che il Web Service mette a disposizione;
- i vincoli del servizio ovvero il protocollo di comunicazione e il formato dei messaggi accettati in input e restituiti in output;
- l'URI⁵ del Web Service ovvero l'“indirizzo” al quale risponde.

A.2 SOAP

Il Simple Object Access Protocol è un protocollo “leggero” per lo scambio dei messaggi tra sistemi software. SOAP può operare su diversi protocolli di rete ma il più usato e l'unico standardizzato dal W3C è HTTP.[12] I messaggi SOAP sono formattati in XML e hanno una struttura header/body:

- **header** (opzionale): contiene informazioni sul tipo di comunicazione come routing e sicurezza;
- **body** (obbligatorio): contiene il messaggio vero e proprio (payload).

A.3 UDDI

L'Universal Description Discovery and Integration è un registry (database ordinato e indicizzato) che permette la pubblicazione da parte delle aziende di dati e servizi offerti.[13]

Un record UDDI consta di:

- **White pages**: contatti dell'azienda che offre uno o più servizi

⁵URI: Universal Resource Identifier

- **Yellow pages:** categorizzazione dei servizi offerti dall'azienda;
- **Green pages:** informazioni sui servizi offerti dall'azienda.

Appendice B

iOS

iOS è il sistema operativo sviluppato da Apple per i propri dispositivi mobili (iPhone, iPod Touch, iPad) che come Mac OS X è un sistema proprietario della famiglia UNIX basato su un microkernel Open Source (Darwin OS).[14] L'interfaccia utente di iOS si basa sulla manipolazione diretta attraverso l'utilizzo di *gestures* multi-touch e la fluidità e immediatezza delle risposte alle azioni dell'utilizzatore.

Al contrario di Windows CE e Android, la licenza di iOS non ne permette l'installazione su hardware di terze parti.

Attualmente l'ultima versione iOS è la 5.1.1. A partire dalla versione 2.0, attraverso un'opportuna SDK¹ fu introdotta la possibilità di installare applicazioni di sviluppatori di terze parti.

B.1 iOS SDK

iOS SDK è un insieme di strumenti e framework che consentono lo sviluppo di applicazioni iOS da distribuire attraverso il negozio **App Store**. Il linguaggio con cui vengono scritte queste applicazioni è Objective-C, linguaggio di programmazione ad oggetti ad alto livello.

¹SDK: Software Development Kit

Tra i vari tool che iOS SDK mette a disposizione agli sviluppatori troviamo:

- **simulatore iOS:** macchina virtuale che emula il funzionamento di un dispositivo Apple e consente di testare il funzionamento di un'applicazione simulando anche alcune funzionalità avanzate come per esempio il comportamento del GPS e degli accelerometri dei dispositivi;
- **XCode:** IDE di riferimento per la scrittura del codice delle applicazioni e per la definizione della loro interfaccia grafica. Tra le varie funzioni, peraltro tipiche in molti altri ambienti di sviluppo, è opportuno un approfondimento sullo strumento Storyboards introdotto dalla versione di XCode 4.2 per iOS 5.0 o superiore.

B.2 Storyboards

Lo Storyboard è uno strumento che va a potenziare le possibilità di progettazione dell'interfaccia grafica rispetto a quanto disponibile in precedenza. Grazie a questo strumento si possono definire tutte le viste (schermate) dell'applicazione, la sequenza logica di navigazione da parte dell'utente e la modalità di presentazione delle viste stesse (per esempio tutto schermo o come popup modale). Tutto ciò è reso possibile grazie all'interfaccia di questo strumento che limita al minimo la scrittura di codice per la creazione e personalizzazione degli oggetti grafici come pulsanti, componenti dinamici (tabelle, pickerView) e sottoviste.

Vengono altresì semplificate altre funzionalità come ad esempio la gestione automatica di un sistema di navigazione tra le viste (NavigationController) e la gestione del TabBarController, la barra inferiore comune a tutte le viste. I comportamenti specifici delle singole viste, successivamente, vanno definiti nel codice delle classi di tipo ViewController seguendo il paradigma MVC.

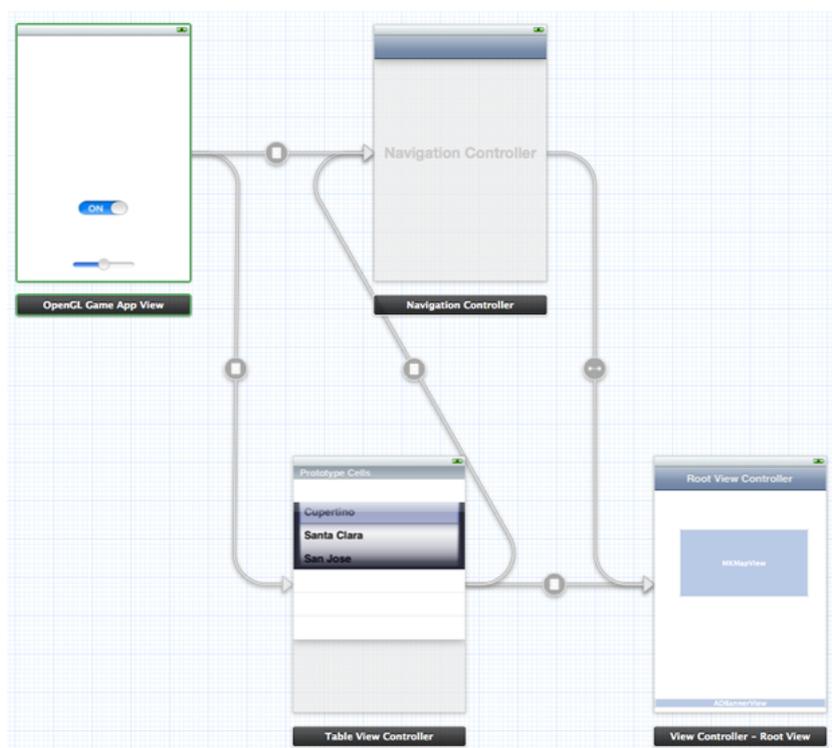


Figura B.1: Storyboard

B.3 App Store

App Store è il marketplace virtuale che permette agli utenti in possesso di un account **Apple ID** la ricerca e l'installazione di applicazioni sul proprio dispositivo iOS. Le applicazioni possono essere gratuite o pagamento. Questo è l'unico market abilitato per il download di applicazioni consumer a meno che i dispositivi non abbiano subito modifiche software non autorizzate dalla licenza d'uso Apple.

Questo deriva da un processo di approvazione, da parte di Apple, di ogni singola applicazione sviluppata da sviluppatori esterni. Il processo prevede un test di sicurezza, un debugging e un controllo di qualità generale dell'applicazione. Tutto ciò è volto a minimizzare la possibilità di installazione di software malevolo e a mantenere un elevato livello qualitativo della piatta-

forma. Tuttavia questa strategia è stata spesso criticata per alcuni episodi di mancanza di trasparenza nelle motivazioni delle mancate autorizzazioni e per la limitata scelta che hanno gli utenti finali nello scegliere applicazioni e marketplace da utilizzare.

Per le applicazioni di tipo business è prevista la distribuzione in-house, cioè limitata ai dispositivi in dotazione ai dipendenti di un'azienda. In questo modo applicazioni custom sviluppate internamente vengono distribuite ai soli dipendenti interessati e non al pubblico mondiale attraverso App Store.

Relativamente a questo tipo di distribuzione delle applicazioni, nel 2010 SAP ha acquisito la software house Sybase che vanta nel suo portafoglio prodotti anche di Afaria, un software dedicato appositamente alla gestione multipattaforma della flotta di dispositivi mobili e alla distribuzione diretta di applicazioni ai device.

B.4 Protocols e Delegation

Una funzionalità molto usata nell'ambito dello sviluppo di applicazioni iOS è l'utilizzo dei protocolli e del meccanismo della delegazione.[15] Il concetto di protocollo prevede la dichiarazione di metodi che possono (o devono) essere implementati da una qualsiasi altra classe. Risultano utili principalmente in tre situazioni:

- per dichiarare l'interfaccia di un oggetto mantenendo nascosta la sua classe;
- per catturare similarità tra oggetti che non sono gerarchicamente collegati;
- per dichiarare metodi che dovranno essere implementati da altre classi.

Spesso i protocolli vengono utilizzati nella programmazione iOS per dar forma al meccanismo della delegation. Con delegation si intende proprio la situazione in cui una classe delega ad un'altra l'implementazione di uno o più

metodi. La classe delegante definisce, nella propria interfaccia, il protocollo con i metodi da delegare e un riferimento generico alla classe delegata. La specifica classe delegata si dovrà conformare al protocollo e implementare i metodi che lo compongono.

Esempio

Supponiamo di avere una classe `Foo` in cui viene dichiarato un protocollo `<pro>` che racchiude `metodoAlfa` e `metodoBeta`:

```
@protocol <pro>

    -(void)metodoAlfa;

    -(void)metodoBeta;

@end
```

Questi metodi devono essere implementati dalle altre classi. Inoltre la classe `Foo` presenta tra le proprie variabili di istanza la variabile `delegate` di tipo `id` (cioè generico, equivalente alla classe `Object` di Java) la quale come unica caratteristica ha quella di dover adottare il protocollo `<pro>`

```
id <pro> delegate;
```

Questi metodi verranno implementati in qualsiasi classe che adotta il protocollo `<pro>`.

Supponiamo che la classe `Bar` contenga un riferimento alla classe `Foo`:

```
Foo myFooObject;
```

Ad un certo punto si avrà la valorizzazione di `delegate`:

```
myFooObject.delegate = self; //l'istanza attuale della classe Bar è il delegate di Foo
```

Nell'implementazione della classe **Bar** saranno presenti le implementazioni dei metodi del protocollo:

```
-(void)metodoAlfa
...
NSLog(@"Hello from metodoAlfa")

-(void)metodoBeta
...
NSLog(@"Hello from metodoBeta")
```

Tutte le classi che adottano il protocollo <pro> dovranno implementare i due metodi.

Bibliografia

- [1] <http://www.businessdictionary.com/>
- [2] <http://www.aberdeen.com/Aberdeen-Library/5374/RA-mobile-business-intelligence.aspx>
- [3] <http://www.sap.com/solutions/sapbusinessobjects/large/enterprise-performance-management/planningandconsolidation/index.epx>
- [4] <http://www.sap.com/italy/about/newsroom/index.epx?pressid=11096>
- [5] <http://www.sap.com/italy/about/newsroom/index.epx?pressid=17238>
- [6] <http://www.sap.com/italy/solutions/mobility/sybase-unwired-platform/index.epx>
- [7] <http://www.sap.com/italy/platform/in-memory-computing/index.epx>
- [8] <http://www.sap.com/italy/about/newsroom/index.epx?pressid=14520>
- [9] <http://www.sap.com/italy/about/newsroom/index.epx?pressid=18115>
- [10] http://en.wikipedia.org/wiki/Web_service
- [11] http://en.wikipedia.org/wiki/Web_Services_Description_Language
- [12] <http://en.wikipedia.org/wiki/SOAP>
- [13] http://en.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration

- [14] <http://en.wikipedia.org/wiki/IOS>
- [15] <https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Chapters/ocProtocols.html>

Ringraziamenti

Ringrazio prima di tutto il mio correlatore Marco che mi ha seguito in tutto il mio lavoro di tesi e Nicholas per la parte client di iTwin BPC.

Alla mia responsabile di lavoro in Twinergy, Benedetta, dico grazie perché è stata sempre comprensiva in tutto questo periodo in cui ho lavorato e studiato.

Ringrazio tanto la mia famiglia: i miei genitori che mi hanno sempre sostenuto con tutto l'amore, l'entusiasmo e la forza possibili e la mia nonna, per l'affetto immenso che mi ha donato in tutti questi anni.

Un grazie speciale va al mio ragazzo, Marco, che ha sopportato pazientemente tutti i miei sbalzi di umore e le nevrosi, ma che soprattutto mi ha sempre aiutato e sostenuto in tutto quello che ho fatto fino ad ora. Grazie per la fiducia, la comprensione e l'Amore.

Infine ringrazio tutti i miei colleghi di università per tutte le giornate di lezione e di studio passate a ridere e scherzare, nelle pause di studio ovviamente.

E ancora tutti gli amici che mi hanno seguito, da vicino e da lontano, durante l'intero corso di studi.