

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

CAMPUS DI CESENA

SCUOLA DI SCIENZE

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

SchoolAssistant

Relazione finale in

MOBILE WEB DESIGN

Relatore

Dott. MIRKO RAVAIOLI

Presentata da

DIEGO CAVALLETTI

III SESSIONE

ANNO ACCADEMICO 2012/2013

Introduzione

Internet è una rete mondiale di reti di computer ad accesso pubblico, attualmente rappresentante il principale mezzo di comunicazione di massa, che offre all'utente una vasta serie di contenuti potenzialmente informativi e servizi.

L'avvento e la diffusione di Internet e dei suoi servizi hanno rappresentato una vera e propria rivoluzione tecnologica e sociologica dagli inizi degli anni novanta (assieme ad altre invenzioni quali i telefoni cellulari e il GPS) nonché uno dei motori dello sviluppo economico mondiale nell'ambito dell'Information and Communication Technology (ICT) e oltre.

Internet garantisce l'accesso ad informazioni e servizi (gratuiti e non) e con l'aiuto della diffusione degli smartphone è ormai presente nella vita della maggior parte delle persone diventando un ottimo strumento da utilizzare anche nel lavoro.

Negli ultimi anni il numero di applicazioni presenti per smartphone è cresciuto in modo esponenziale, un chiaro segnale della enorme utilità nella vita quotidiana.

Le applicazioni attualmente disponibili sono in grado di svolgere i compiti più dissimili, dal gioco al lavoro, dal social alla salute personale.

Questa tesi ha come obiettivo la realizzazione di un' applicazione per smartphones dotati di sistema operativo Android da inserire nel settore dell'istruzione.

Nello specifico l'applicazione "SchoolAssistant" è stata sviluppata nell'ottica di offrire ai docenti uno strumento che metta a disposizione un discreto numero di funzionalità per migliorare la qualità del lavoro svolto, ai genitori la possibilità di controllare l'andamento scolastico dei propri figli, agli studenti di avere sempre sotto controllo le comunicazioni e i voti ricevuti durante l'anno scolastico e alla scuola un servizio in più da offrire ai suoi iscritti.

1. Storia degli smartphone

L'idea di creare dei dispositivi che unissero la telefonia all'utilizzo del computer risale al 1973, ma le prime vendite di tali dispositivi iniziarono solo nel 1993. Il termine "smartphone" invece è apparso dal 1997 quando la Ericsson descrisse il suo GS 88 "Penelope" uno **SmartPhone**.

La storia degli smartphone per il mercato di massa comincia con il successo dell'iPhone di Apple, ma in realtà il primo modello di telefono con funzioni allargate è datato 1993, costruito dalla IBM.



Illustrazione 1 : Simon IBM

In meno di 20 anni si è passati da rudimentali apparecchi dall'aria preistorica ai moderni gioielli della tecnologia odierna, forti di processori con potenza di calcolo paragonabile ai migliori supercomputer dei primi anni 90.

1.1 Simon : il primo “smartphone”

Simon è stato a tutti gli effetti il primo vero tentativo dell'industria tecnologica di creare un telefono con funzioni di voce e dati. Simon era in grado di telefonare, spedire Fax e fungere da agenda PDA. Come tutti gli smartphone che si rispettano era, ovviamente, touchscreen: un tocco che portava direttamente nel futuro.

Era un prodotto rivolto alla sola clientela business, per i costi proibitivi delle comunicazioni e anche per i necessari 899 \$ di acquisto. Ma quello che lo rendeva più caro di tutto era la mentalità di quegli anni, la quale difficilmente concepiva la necessità di un telefono tutto fare. La comunicazione viaggiava ancora per il canale classico, quello della televisione. Anche se tutte le Università si stavano collegando in Rete, e cominciavano a fiorire le prime BBS.

Le stesse compagnie telefoniche non vedevano in un oggetto simile un prodotto di massa, e quindi lo avevano condannato ad un uso rivolto a pochissimi fortunati. Sol dopo 15 anni le compagnie intuiscono il nuovo business della comunicazione globale su piattaforma mobile, proprio con l'arrivo del “melafonino”.

1.2 Palm Pilot : il precursore dell'avvento degli smartphone

Il Palm Pilot non era un vero smartphone, perché non telefonava. Eppure la sua diffusione negli ambienti di lavoro fu una vera scossa per gli utenti, che potevano portare con sé sempre l'intera rubrica telefonica, prendere appunti e segnare ogni tipo di appuntamento nel calendario interattivo. Costava 300 \$, aveva una memoria di 128KB e un processore di 16MHz mancante solo dell'opzione voce, però è stato il primo “terremoto” nell'ambiente palmari/agende elettroniche intelligenti.

1.3 Nokia 9110 Communicator : l'esempio da seguire

Nel 1998 arriva il Nokia 9110 Communicator (dal costo proibitivo) con le sembianze di un moderno smartphone, lo schermo in scala di grigi, non era però in grado di navigare sul Web, tuttavia è stato preso come esempio negli anni successivi. Per questo motivo Nokia è diventata, in quegli anni, il leader mondiale del settore.

1.4 BlackBerry 5810 : Il vero smartphone

Alla fine degli anni '90, la canadese Research in Motion, conosciuta soprattutto per i suoi “cercapersone” (utilizzati da decine di milioni di utenti nel mondo) progettava un telefono capace di navigare e di leggere le e-mail in tempo reale: una vera rivoluzione, un big bang nel campo della telefonia mobile che avvierà l’era moderna degli smartphone. Il BlackBerry 5810 è stato a tutti gli effetti il primo vero smartphone della storia e i canadesi sono gli inventori del telefonino computerizzato.

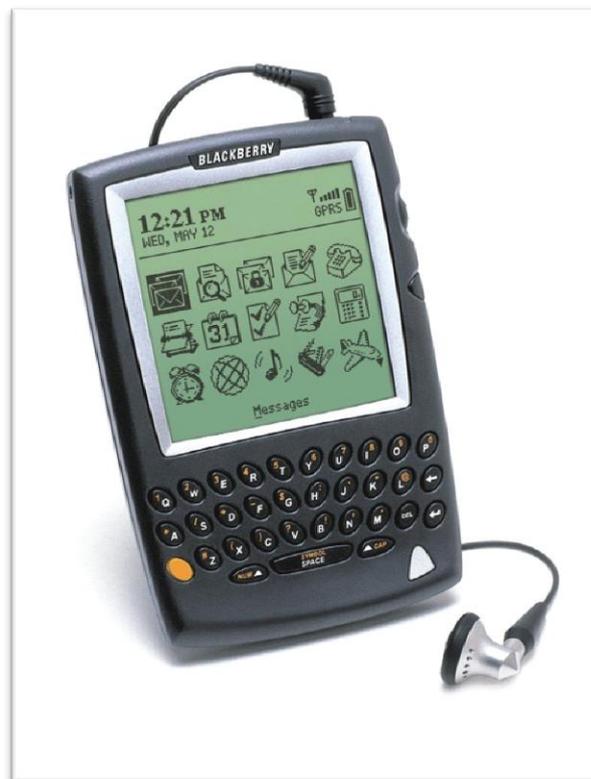


Illustrazione 2: BlackBerry 5810 - Il primo smartphone

1.5 Apple Iphone 1 : lo smartphone per tutti

Il 2007 è l'anno di Apple: Cupertino sfida il mercato mondiale col prodotto concepito proprio secondo la filosofia "user friendly". Si tratta dell'apparecchio più semplice sul mercato, il migliore in qualità. L'iPhone diventa subito il prodotto di riferimento per tutti. Una moda trasversale che resiste intatta dopo 4 anni.



Illustrazione 3 : Apple Iphone 1

1.6 Gli smartphone al giorno d'oggi

Android è nato alla fine del 2007 ed è il sistema di riferimento per la filosofia estremista dell'open source ad ogni costo. Il suo successo è numericamente straordinario. Sbaraglia in poco tempo BlackBerry, Windows Phone e soprattutto Symbian. Android è la vera alternativa al telefono targato Apple.

Il Motorola Droid rappresenta il primo vero grande successo di Android, successo inteso nel numero di apparecchi venduti. Il Droid fu venduto in più di 1 milione di unità nel primo mese, supportato anche dalla potente rete di Verizon.

HTC e Samsung spingono ulteriormente lo sviluppo dei propri telefoni adattandoli al sistema operativo di Google. La rete dati è la velocissima 4G, siamo ormai ai giorni nostri.



Illustrazione 4 : Gli smartphone al giorno d'oggi

In appena 18 anni i cellulari sono diventati sottili, veloci e intelligenti, la comunicazione ormai è solo su canali dati, ed il futuro ormai è proprio questo. Il mercato globale arriva nel 2011 a un valore del 2% del PIL mondiale. I computer verranno unificati con i dispositivi mobili, il cloud diventerà il collante di ogni apparecchio digitale.

2. Il cuore degli smartphone : Il sistema operativo

Trattando l'argomento smartphone è doveroso descrivere ciò che permette al dispositivo di eseguire tutte le funzioni che il proprio hardware permette.

Un sistema operativo (abbreviato in SO, in lingua inglese OS, "operating system"), in informatica, è un insieme di componenti software, che consente l'utilizzo di apparecchiature informatiche da parte di un utente.

Esso garantisce l'operatività di base del dispositivo, coordinando e gestendo le risorse hardware di processamento (processore) e memorizzazione (memoria primaria), le periferiche, le risorse/attività software (processi) e facendo da interfaccia con l'utente, senza il quale quindi non sarebbe possibile l'utilizzo dello smartphone stesso e delle applicazioni su di esso installate.

È dunque un componente essenziale che funge da "intermediario" tra l'utente e il dispositivo ed è inoltre una base a cui si appoggiano gli altri software che dovranno essere progettati e realizzati per essere riconosciuti e supportati da quel particolare sistema operativo.

Nell'ambito dei dispositivi mobili il sistema operativo affronta problematiche, legate alla natura stessa dello smartphone, più critiche rispetto ad un desktop o un laptop: limitatezza delle risorse (memoria, CPU), assenza di alimentazione esterna, differenti tecnologie per l'accesso a Internet (WiFi, GPRS, HSDPA), nuovi metodi d'immissione (touchscreen, minitastiere) e ridotte dimensioni del display.

I SO di maggioranza nell'ambito mobile sono : Android, Windows Phone, Symbian, Blackberry e IOS.

2.1 Android : il SO mobile più diffuso

Con appena 10 anni di vita sulle spalle, Android è il sistema operativo più utilizzato per smartphone e tablet.

Tutto inizia più o meno 10 anni fa, quando Andy Rubin, Rich Miner, Nick Sears e Chris White fondano Android Inc. , piccola società con grandi ambizioni. Inizialmente Android doveva servire da sistema operativo per fotocamere digitali, in modo che gli utenti potessero installare nuove app e aggiungere nuove funzionalità ai loro dispositivi.

Nel 2005 sul palcoscenico Android fa la sua apparizione anche Google, quello che stava per diventare un gigante del web e dell'hi-tech inizia a finanziare il progetto di Rubin e degli altri e, dopo pochi mesi, ne prende il controllo.

Android si caratterizza per la struttura open source e il suo basarsi su kernel Linux. La caratteristica open source ed il tipo di licenza permette di modificare e distribuire liberamente il codice sorgente.

Android dispone di una vasta comunità di sviluppatori che realizzano applicazioni con l'obiettivo di aumentare le funzionalità dei dispositivi.

E' stato progettato principalmente per smartphone e tablet, ma il carattere aperto e personalizzabile del sistema operativo permette il suo utilizzo anche su altri dispositivi elettronici tra cui portatili e netbook, smartbook, eBook reader, fotocamere e smart TV (Google TV). Il mercato delle "smart things" è cresciuto in maniera notevole in questi ultimi periodi a tal punto da stimolare la creatività delle persone. Un esempio è lo Smartwatch dotato di sistema operativo Android in versione "light", cuffie, lettori auto CD e DVD, occhiali intelligenti (Project Glass o google glass), frigoriferi, sistemi di navigazione satellitare per veicoli, sistemi di automazione per la casa, console di gioco, specchi, telecamere, lettori MP3/MP4 e tapis roulant.

Ad oggi sono state sviluppate diverse versioni del sistema operativo Android, ricordandone alcune: Gingerbread (2.3/2.4), Honeycomb (3.0), Ice Cream Sandwich (4.0) , Jelly Bean (4.1) e KitKat (4.4).

Android dispone di un market (chiamato Google Play Store) in cui è possibile scaricare un quantitativo sempre maggiore di applicazioni.

2.2 Windows Phone : un SO in crescita

Windows Phone è il sistema operativo per smartphone di Microsoft. Si rivolge al mercato consumer invece che al mercato enterprise come il suo predecessore, abbandonando alcune caratteristiche in dotazione a Windows Mobile.

È radicalmente diverso da tutte le precedenti versioni di quest'ultimo, con le quali è incompatibile. Consente l'integrazione degli account di social network quali Facebook e Twitter, e possiede una nuova versione di Internet Explorer Mobile basata su quella per desktop. Windows Phone contiene un'edizione di Office Mobile, con Word, Excel, Powerpoint, OneNote e SharePoint.

L'11 febbraio 2011 durante una conferenza stampa a Londra, Steve Ballmer e l'amministratore delegato di Nokia Stephen Elop annunciarono una partnership tra le loro aziende in base alla quale Windows Phone sarebbe diventato il principale sistema operativo dei futuri smartphone Nokia. L'evento era principalmente focalizzato sulla creazione di "un nuovo ecosistema globale", affiancando competitor come Android e iOS, parlando di "una gara a tre competitor". In base all'accordo, Bing sarebbe stato il motore di ricerca per i telefoni Nokia, portando ad un'integrazione tra Nokia Maps e Bing Maps così come tra l'OVI di Nokia e il Windows Phone Marketplace.

I primi smartphone Nokia a montare il sistema operativo della software house di Redmond furono i Nokia Lumia 800 e 710, presentati ad ottobre al Nokia World 2011

Il problema principale di Windows Phone è la presenza di un minore numero di applicazioni presenti nel suo Store rispetto ai principali concorrenti, tuttavia il gap si sta progressivamente riducendo.

2.3 Symbian OS : un progetto abbandonato

Symbian OS è un sistema operativo per dispositivi mobili prodotto da Symbian Foundation. Lo sviluppo di Symbian OS è attualmente supportato da Accenture, ma, dopo l'abbandono da parte di Nokia, nessun hardware lo supporta più.

Il Symbian OS è l'erede del sistema operativo EPOC, nella sua incarnazione EPOC32, creato dalla Psion alla fine degli anni novanta per la sua linea di palmari. La sua nascita risale al giugno del 1998 con la creazione della compagnia indipendente Symbian Limited nata dalla cooperazione di diverse compagnie telefoniche e dalla stessa Psion. Il primo telefono mobile ad utilizzare questo sistema operativo fu il Sony Ericsson modello P800 e i Nokia 9210 e 7650. L'ultima versione disponibile del sistema operativo è la 5.0.

Nel giugno del 2008 Nokia ha comunicato l'intenzione di rilevare le quote azionarie delle altre società al fine di divenire l'unico proprietario del sistema operativo, e una volta completata l'acquisizione di renderlo open source con la creazione di Symbian Foundation: essa sarà formata dai vecchi proprietari e aperta ad altri produttori, e si dovrà occupare di unificare tutte le interfacce in una nuova release del sistema operativo e gestirne l'apertura agli sviluppatori esterni. Symbian dal 4 febbraio 2010 è diventato un sistema operativo libero. La documentazione relativa alle API è disponibile pubblicamente, quindi chiunque ha la possibilità di sviluppare software per Symbian. L'11 febbraio 2011 Nokia e il suo nuovo CEO Stephen Elop hanno scelto di adottare sul proprio hardware il nuovo sistema operativo di Microsoft, Windows Phone 7, come "piattaforma principale" affiancando la piattaforma Symbian. Il 5 aprile 2011 Nokia ha annunciato un cambiamento nei criteri necessari per contribuire al progetto, trasformandolo da un progetto libero a codice aperto a un progetto dove solo le aziende potranno collaborare al suo sviluppo. Il 27 aprile 2011 Nokia ha annunciato l'esternalizzazione totale dello sviluppo della piattaforma Symbian alla società Accenture, già da tempo partner strategico della società finlandese.

Il 24 gennaio 2013, Nokia ha comunicato di non voler più utilizzare Symbian per i propri dispositivi. L'11 giugno 2013, Nokia ha comunicato di terminare la produzione di dispositivi dotati di Symbian.

Dal 1° gennaio 2014 Nokia ha chiuso lo store Symbian non accettando più nuovi contenuti né aggiornamenti per le applicazioni già pubblicate.

2.4 Blackberry

BlackBerry OS è un sistema operativo mobile proprietario, sviluppato da Research In Motion per la sua linea di smartphone BlackBerry. Il sistema operativo fornisce multitasking e supporta dispositivi di input specializzati che sono stati adottati da RIM, in particolare la rotellina (trackball) e più recentemente il trackpad e touchscreen.

La piattaforma BlackBerry è forse più nota per il suo supporto nativo della posta elettronica aziendale attraverso MIDP 1.0 (Mobile Information Device Profile) e più recentemente, un sottoinsieme di MIDP 2.0, che consente di completare l'attivazione wireless e la sincronizzazione con Microsoft Exchange, Lotus Domino o Novell GroupWise, per la sincronizzazione di e-mail, calendario, attività, note e contatti (necessità l'attivazione del BlackBerry Enterprise Server). Il sistema operativo supporta anche il WAP 1.x e WAP 2.0. Gli aggiornamenti del sistema operativo sono messi a disposizione dai diversi operatori di telefonia mobile (in quanto lo riadattano alle loro reti ed esigenze) che supportano il BlackBerry tramite il caricamento del software (utilizzando l'applicazione da PC), anche in OTASL, quindi senza dover connettere il dispositivo al personal computer (tramite la funzione aggiornamento wireless), l'unico dispositivo BlackBerry che non ha gli aggiornamenti vincolati dall'operatore telefonico è il playbook. Sviluppatori di terze parti sono in grado di scrivere software (applicazioni) utilizzando le classi API disponibili per BlackBerry OS, anche se le applicazioni che fanno uso di determinate funzionalità devono essere firmate digitalmente, queste applicazioni possono essere aggiornate direttamente dal telefono tramite l'applicazione AppWorld.

2.5 iOS

Al momento della sua nascita, collegata con quella del primo iPhone, non era come oggi ma molto più introverso e con poche funzionalità, iPhone OS 1 era veramente basic. In pochi ne compresero subito le potenzialità, anche perchè quando venne lanciato, Symbian (il sistema operativo che Nokia utilizzava sui propri smartphone) era ancora sulla cresta dell'onda. Non potendo installare applicazioni di terzi, iPhone OS 1 era praticamente da scegliere e da tenere così come lo si era comprato.

Una delle prime critiche mosse all'iPhone dopo il suo lancio sul mercato fu la sua estrema chiusura. Principalmente le critiche degli utenti furono tre :

- Impossibilità di mandare MMS (molto in voga nel 2007)
- l'impossibilità di scambiare file tramite Bluetooth
- assenza del supporto ad Adobe Flash.

Il pubblico vide queste tre impossibilità come estremamente negative. Oggi quasi nessuno scambia più file tramite MMS, sostituendo questa funzione con applicazioni più efficienti. Per quanto riguarda Adobe Flash, dopo aver ricevuto moltissime critiche da altri brand, il tempo ha dato ragione ad Apple e anche altri OS quali Android hanno deciso di togliere il supporto a questa tecnologia in favore dell'HTML5.

Con l'introduzione dell'Apple Store nasce il motto "esiste un'applicazione per tutto". Le assenze sono effettivamente poche, si trovano App sia gratis sia a pagamento in base alle richieste dell'utente.

3. La tecnologia nell'istruzione

La tecnologia è ormai entrata a far parte anche del settore dell'istruzione integrandosi ai metodi di apprendimento classici e offrendo agli studenti nuovi mezzi per raggiungere gli obiettivi previsti dal percorso di studio.

L'avvento dei primi computer ha sicuramente contribuito ad introdurre nell'istruzione un nuovo insegnamento: l'informatica.

L'informatica è la scienza che ha per oggetto lo studio dei fondamenti teorici dell'informazione, della sua computazione a livello logico e delle tecniche pratiche per la sua implementazione e applicazione in sistemi elettronici automatizzati detti appunto sistemi informatici .

Le applicazioni in ambito scolastico dell'informatica coprono gli aspetti più vari, dal semplice studio della materia stessa, avendo a disposizione uno strumento interattivo, alla possibilità di fruire di contenuti multimediali.

Negli ultimi anni, con il crescente sviluppo di internet e l'obiettivo di offrire ai propri iscritti maggiori servizi, le scuole hanno sempre più spesso a disposizione un sito internet da cui consultare ogni tipo di informazione sulla struttura, gli insegnanti ed i corsi disponibili.

Tra i servizi offerti dai siti internet scolastici si trova sempre più spesso il registro elettronico.

Il registro scolastico è un strumento utilizzato dal professore per annotare le presenze degli alunni, i contenuti svolti, le valutazioni e annotazioni varie. Esistono diversi tipi di registri scolastici:

Il Registro di classe è utilizzato per depositare le firme dei professori, le assenze, le giustificazioni degli alunni, i compiti assegnati, i lavori svolti e i dati generali degli alunni. La sua funzione è anche quella di annotare i provvedimenti disciplinari (note disciplinari, sospensioni). Insieme al Giornale del Professore, attesta, inoltre, la presenza dei docenti. Il Registro di classe è visionabile da tutti gli alunni, ma può essere modificato solo dai docenti o dal Dirigente scolastico.

Giornale del Professore, conosciuto anche come Registro del Professore, è un documento cartaceo dato in dotazione ad ogni singolo insegnante nel quale si annotano assenze e valutazioni relative ad ogni alunno nella propria materia. Il Giornale del Professore

contiene solitamente un'aletta pieghevole con i nomi degli alunni, delle griglie per le assenze, delle pagine per la programmazione, delle pagine per l' annotazione degli argomenti delle lezioni e delle pagine per la valutazione degli alunni. Talvolta, a fianco alla pagina della valutazione del primo alunno, può essere presente una seconda aletta pieghevole con gli Obiettivi Formativi/Obiettivi Specifici di Apprendimento (O.F./O.S.A.) per le rispettive Unità di Apprendimento (U.A./U.d.A.).

Il decreto legge n° 95/2012 introduce per le Istituzioni scolastiche ed i docenti l'obbligo di dotarsi di registro elettronico a decorrere dall'anno scolastico 2012-2013.

Il "registro elettronico" è il registro "virtuale" che potrà essere consultato on-line dai genitori dopo aver ottenuto dalla propria scuola i dati di accesso. Le famiglie possono tenersi informate in tempo reale sulla regolarità della frequenza dei propri figli, sulle assenze, ritardi e uscite in anticipo oltre che sui risultati nelle varie discipline di studio. Il sistema riproduce in ogni minimo dettaglio i tradizionali registri scolastici e utilizzando la rete Internet.

- Rileva quotidianamente le assenze, i ritardi, le uscite anticipate e le giustificazioni degli allievi.
- Informa sulle valutazioni e comportamenti degli allievi.

I vantaggi sono molteplici in quanto rende possibile:

- accedere ai servizi senza vincoli temporali e spaziali.
- garantire la trasparenza dell'azione didattica-educativa dichiarata nel "contratto formativo" tra alunni, docenti e genitori.
- monitorare continuamente l'andamento scolastico dell'alunno
- programmare tempestivamente interventi mirati di recupero e sostegno.

Tali servizi sono fruibili solamente visitando il sito web della propria scuola.

Il progetto descritto in questa tesi consentirà di rendere comodamente accessibile il registro elettronico direttamente dalle "tasche" degli utenti sotto forma di applicazione scaricabile gratuitamente dal Play Store.

Le funzioni disponibili scelte per l'applicazione sono quelle fondamentali scelte tra quelle presenti online.

4. Progettazione

In questo capitolo si introducono le fasi iniziali per lo sviluppo dell'applicazione "SchoolAssistant", nello specifico si tratta l'analisi che è a sua volta suddivisa in più step.

- Analisi dell'ambiente applicativo e degli utenti finali
- Analisi delle funzionalità richieste
- Analisi strutturale dell'applicazione
- Analisi dell'interfaccia utente

4.1 Analisi dell'ambiente applicativo e degli utenti finali

Il primo aspetto su cui soffermarsi durante lo sviluppo è avere una chiara idea dell'ambiente in cui l'applicazione andrà ad inserirsi (un edificio scolastico in questo caso) e degli utenti finali che dovranno utilizzarla.

Come detto in precedenza la tecnologia nell'istruzione non è sicuramente cosa nuova, d'altra parte la possibilità di avere sempre a disposizione uno smartphone è sicuramente più recente.

Considerando gli aspetti culturali del nostro Paese e osservando in modo oggettivo l'istruzione si può assumere che l'utente finale medio non ha una gran dimestichezza con la più recente tecnologia, soprattutto con gli smartphone, si rende quindi necessario uno strumento estremamente intuitivo, funzionale e facile da utilizzare con opzioni a portata di "touch".

4.2 Analisi delle funzionalità richieste

L'applicazione è strutturata per essere utilizzata da 5 figure principali elencate per quantità di privilegi decrescente:

- Amministratore
- Segretario/a
- Docenti
- Genitore
- Studente

L'applicazione è temporaneamente collocata in un servizio di hosting online gratuito (www.altervista.org) che riserva ai suoi utenti uno spazio di alcune centinaia di MB sul quale posizionare i file necessari.

Le comunicazioni saranno quindi strettamente legate alla presenza di connettività da parte dell'utente finale in quanto tutti i dati risiederanno nel database online e non in locale sul dispositivo.

Non sono previste funzioni eseguibili dall'utente senza prima autenticarsi al sistema online, verrà fornita un' interfaccia di login e registrazione affinché sia possibile per l'utente utilizzare l'applicazione. I nuovi utenti così creati verranno assegnati di default ad un livello fittizio paragonabile a quello di uno studente. Sarà poi compito dell'amministratore o della segreteria assegnare i livelli di utenza opportuni.

4.2.1 Amministratore di sistema

L'amministratore è il primo utente registrato e dispone del più alto numero di privilegi all'interno dell'applicazione; si suppone che gli amministratori siano in numero estremamente limitato in quanto possono modificare parti sensibili dell'applicazione e potrebbero causarne un funzionamento non corretto, al momento dello sviluppo. Le funzionalità a disposizione sono state limitate alle seguenti:

- Inserimento di un nuovo utente (di qualsiasi livello, anche un altro amministratore)
- Modifica di dati specifici dell'utente (ad esempio la classe per uno studente o un insegnante)
- Inserimento di una nuova struttura scolastica
- Inserimento di una nuova classe
- Visualizzazione di tutti gli utenti registrati

4.2.2 Personale della segreteria

Il personale della segreteria è quel gruppo di utenti che più si avvicina alla figura di amministratore, anch'esso ha la capacità di creare nuove classi e utenti (esclusi gli amministratori).

E' responsabile di inoltrare comunicazioni agli altri utenti nel caso in cui ci siano novità riguardanti lo svolgimento delle attività scolastiche.

4.2.3 Docenti

I docenti sono gli utenti principali dell'applicazione, intorno alla loro figura è costruita l'applicazione in tutte le sue forme. Tramite l'applicazione saranno in grado di seguire i loro studenti e le loro attività nel modo più flessibile, veloce ed intuitivo possibile.

Le funzionalità principali richieste per i docenti sono strettamente legate al concetto di registro cartaceo, spaziano quindi dalla possibilità di eseguire l'appello a quello di inviare comunicazioni direttamente ai genitori nel caso di problemi con gli studenti. Ogni docente ha la possibilità di consultare il registro di personale relativo ad ognuna delle classi a cui è stato assegnato interagendo con tutti gli studenti che ne fanno parte. La funzione più immediata e facilmente accessibile da offrire è sicuramente l'appello rendendo disponibile ovviamente anche la possibilità di consultare le assenze di qualsiasi giorno dell'anno scolastico.

Altrettanto importante è l'opportunità di registrare voti per gli studenti coprendo così quelle che sono le "funzionalità" del registro cartaceo, la miglioria introdotta dall'applicazione è quella di poter consultare in qualsiasi momento la lista dettagliata di tutti i voti relativi ad ogni singolo studente e la media aritmetica di quest'ultimi. L'ultima funzionalità riguarda il rapporto tra docenti e genitori degli studenti consolidato da un sistema diretto di comunicazione tra le due figure attraverso l'uso dell'applicazione.

4.2.4 Genitori

Rappresentano un gruppo di utenti con funzionalità prevalentemente di controllo, ad essi viene data la possibilità di consultare i voti assegnati nelle varie discipline e di ricevere tutte le annotazioni e comunicazioni riguardanti i propri figli.

Come detto nel precedente paragrafo si crea un canale diretto di comunicazione unidirezionale (ai genitori non è data la possibilità di inviare comunicazioni ai docenti) per informare tempestivamente sulla situazione scolastica dell'alunno.

4.2.5 Studenti

Gli studenti rappresentano l'utente col minor numero di privilegi, le loro possibilità sono limitate alla banale consultazione dei dati riguardanti il loro personale rendimento scolastico e la lettura delle comunicazioni che la segreteria o i docenti potrebbero aver inviato.

4.3 Analisi strutturale dell'applicazione

Terminata l'analisi delle funzionalità si è pensato a come strutturare l'applicazione, in questa fase delicata si prendono decisioni che ne andranno poi ad influenzare profondamente l'essenza stessa.

4.3.1 La scelta del sistema operativo

La prima decisione da prendere parlando di dispositivi *mobile* è senz'altro scegliere per quale sistema operativo sviluppare la propria applicazione.

Le opzioni valide sono : Android,iOS,Windows Phone,Blackberry e la scelta è ricaduta su Android che , come mostra la tabella seguente, è il sistema operativo *mobile* attualmente più diffuso al mondo.

| Top Four Operating Systems, Shipments, and Market Share, Q3 2013 (Units in Millions) | | | | | |
|--|-----------------------|-------------------|-----------------------|-------------------|-----------------------|
| Operating System | 3Q13 Shipment Volumes | 3Q13 Market Share | 3Q12 Shipment Volumes | 3Q12 Market Share | Year-Over-Year Change |
| Android | 211.6 | 81.0% | 139.9 | 74.9% | 51.3% |
| iOS | 33.8 | 12.9% | 26.9 | 14.4% | 25.6% |
| Windows Phone | 9.5 | 3.6% | 3.7 | 2.0% | 156.0% |
| BlackBerry | 4.5 | 1.7% | 7.7 | 4.1% | -41.6% |
| Others | 1.7 | 0.6% | 8.4 | 4.5% | -80.1% |
| Total | 261.1 | 100.0% | 186.7 | 100.0% | 39.9% |

Illustrazione 5: Mobile SO sharing world wide

Tra i vantaggi di tale scelta si ricorda che Android dispone di una grande community formata da utenti che condividono le loro esperienze nell'ambito della programmazione *mobile*, ciò comporta una semplificazione nella reperibilità delle informazioni necessarie allo sviluppo.

Il linguaggio di programmazione utilizzato è Java che viene utilizzato in numerosi altri ambiti contribuendo ad incrementare la quantità di documentazione disponibile.

Il numero di dispositivi su cui viene installato il SO Android come si può evincere dalla tabella è superiore a tutti gli altri, il motivo risiede sicuramente anche nella variabilità del costo di questi prodotti, ricordando questo aspetto possiamo quindi assumere che in un edificio scolastico che si appresta ad utilizzare “SchoolAssistant” sarà più probabile avere a che fare con un dispositivo marcato Android .

Un ultimo aspetto che ha inciso sulla scelta finale è la disponibilità di dispositivi su cui testare l’applicazione, per i motivi sopra elencati resta sempre più agevolato lo sviluppo per Android.

Terminando l’esplicazione dei motivi riguardanti la scelta effettuata chiarendo le motivazioni che hanno portato all’esclusione degli altri SO mobile disponibili.

La scelta non è ricaduta su BlackBerry a causa della sua scarsa diffusione che avrebbe reso l’applicazione poco disponibile in termini di utenza.

Windows Phone è ancora troppo “giovane”, non si hanno a disposizione molti esempi su cui basare le proprie idee ne’ sufficienti guide per chiarire i dubbi e i problemi che sicuramente si presenteranno in fase di sviluppo.

Potrebbe sembrare azzardato escludere anche iOS, elenchiamo quindi gli svantaggi di questo SO:

- Elevato costo dei dispositivi
- Applicazioni sviluppate in Objective-C, un linguaggio di programmazione utilizzato esclusivamente nella produzione di app per iOS
- Community degli sviluppatori estremamente “chiusa” nel voler diffondere le proprie conoscenze ed esperienze
- Scarsa possibilità di personalizzazione nell’aspetto dell’applicazione
- Lunghe procedure e costo elevato per la pubblicazione su Apple Store

Scelto il sistema operativo è stato necessario determinare per quale versione di Android sviluppare l'applicazione, in nostro aiuto tornano nuovamente i dati registrati dal mercato delle vendite:

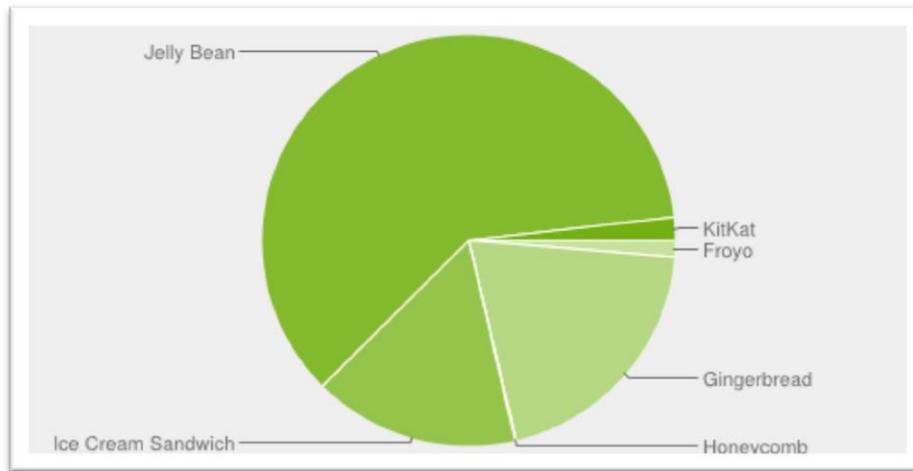


Illustrazione 6 : Android version chart

| Version | Codename | API | Distribution |
|---------------|--------------------|-----|--------------|
| 2.2 | Froyo | 8 | 1.3% |
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 20.0% |
| 3.2 | Honeycomb | 13 | 0.1% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 16.1% |
| 4.1.x | Jelly Bean | 16 | 35.5% |
| 4.2.x | | 17 | 16.3% |
| 4.3 | | 18 | 8.9% |
| 4.4 | KitKat | 19 | 1.8% |

Come possiamo vedere dalla tabella Jelly Bean è la più diffusa, l'applicazione sarà quindi supportata solamente dalla versione 4.1.x in avanti (API 16).

4.3.2 Struttura delle comunicazioni

Seguendo le specifiche precedentemente menzionate notiamo che l'applicazione richiede un continuo scambio di dati tra il dispositivo e uno spazio web propriamente configurato.

Lo spazio web selezionato per questa tesi mette a disposizione un server web apache capace di gestire pagine PHP (acronimo ricorsivo di Hypertext Preprocessor) e un database basato su MySQL.

In linea con i servizi offerti la parte web che si occuperà di elaborare le richieste effettuate dagli utenti tramite l'applicazione saranno scritte in PHP. Saranno le stesse pagine PHP ad interfacciarsi col database permettendo quindi l'accesso a tutti i dati nelle modalità richieste.

La scelta di salvare tutti i dati online è sicuramente intrinseca del tipo di applicazione richiesto. Poiché "SchoolAssistant" è pensata per più strutture e più utenti che interagiscono contemporaneamente col sistema sarebbe stato estremamente negativo propendere per un salvataggio locale dei dati, ciò avrebbe reso difficili le sincronizzazioni necessarie tra i vari utenti.

Ogni operazione effettuata dall'applicazione sarà una richiesta HTTP ad una pagina PHP specifica che produrrà un risultato in formato JSON. Il JSON, acronimo di JavaScript Object Notation, è un formato adatto per lo scambio dei dati in applicazioni client-server. L'applicazione si occuperà poi di gestire la risposta nel modo più opportuno.

4.3.3 Struttura del database

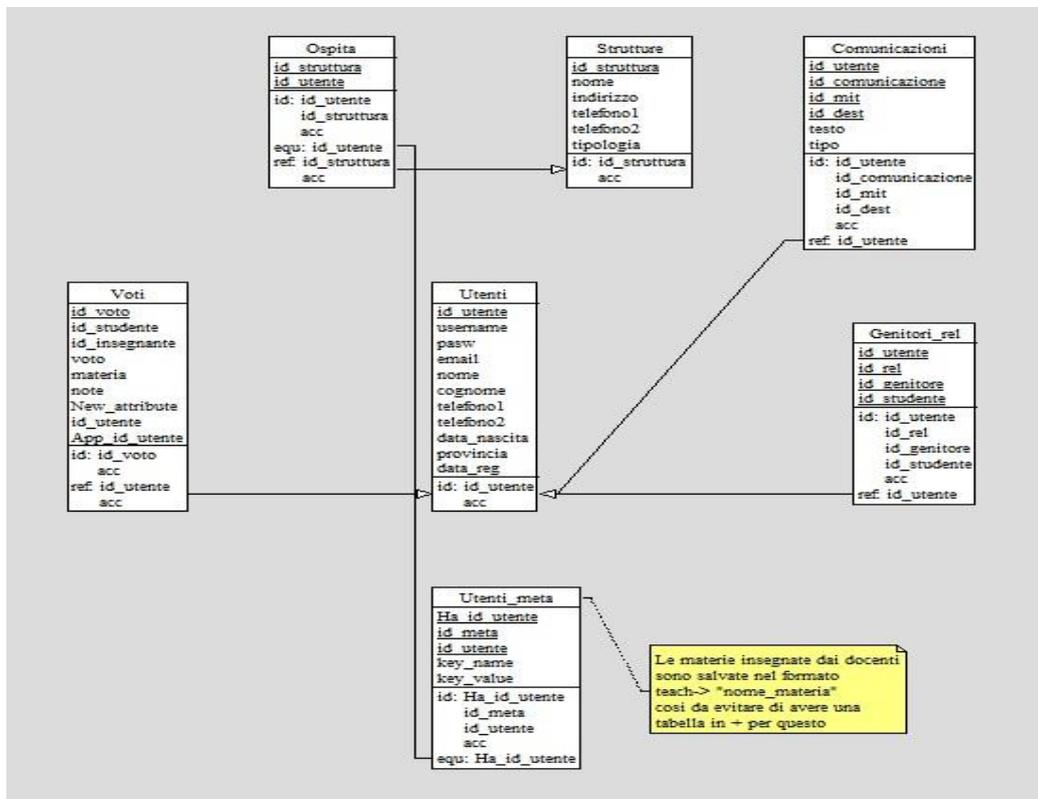
Un database è un archivio dati in cui le informazioni in esso contenute sono strutturate e collegate tra loro secondo un particolare modello logico (relazionale, gerarchico, reticolare o a oggetti) e in modo tale da consentire la gestione/organizzazione efficiente dei dati stessi e l'interfacciamento con le richieste dell'utente attraverso i cosiddetti query language (query di ricerca o interrogazione, inserimento, cancellazione, aggiornamento ecc.) grazie a particolari applicazioni software dedicate (DBMS), basate su un'architettura di tipo client-server. Per descrivere il database utilizzeremo il modello E/R.

Il modello entity-relationship (anche detto modello entità-relazione, modello entità-associazione o modello E-R) è un modello per la rappresentazione concettuale dei dati ad un alto livello di astrazione, formalizzato dal prof. Peter Chen nel 1976 . Viene spesso utilizzato nella prima fase della progettazione di una base di dati in cui è necessario tradurre le informazioni risultanti dall'analisi di un determinato dominio in uno schema concettuale.

Il software utilizzato in questo progetto per rappresentare la struttura del database è DB-Main (strumento con licenza gratuita pensato per aiutare sviluppatori e analisti nella fase di progettazione).

I principali costrutti utilizzati nel modello E/R sono:

- Entità: rappresentano classi di oggetti che hanno caratteristiche in comune. L'istanza di un'entità rappresenta un singolo oggetto appartenente a quell'entità.
- Associazione: rappresentano un legame tra due o più entità.
- Attributo:rappresenta una proprietà che descrive un entità o un'associazione.
- Identificatore: costituisce un sottoinsieme di attributi di un'entità che identifica in maniera univoca ogni istanza della stessa entità, è indicato graficamente come una linea sottostante gli attributi che compongono l'identificatore.



Analizzando la struttura del database a partire dalla tabella base “Struttura” da cui si sviluppa poi l’intero progetto si osserva che è composta dai seguenti campi:

- Id_struttura : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Nome : nome della struttura
- Indirizzo
- Telefono1 : numero di telefono principale
- Telefono2 : numero di telefono secondario
- Tipologia : tipologia del corso di studi (Liceo,Istituto tecnico,professionale,ecc...)

I dati in questa tabella saranno inseriti dall’amministratore di sistema.

Proseguendo con ordine lo schema troviamo la tabella “Ospita” che rappresenta i dati relativi all’associazione tra un determinato utente ed una specifica struttura, tuttavia non è fisicamente presente nel database reale, la motivazione verrà approfondita quando si tratterà la tabella “utenti_meta”.

La tabella principale dell’applicazione è senz’altro “utenti”, in questa tabella verranno registrati tutti gli utenti indipendentemente dal loro livello, è composta dai seguenti campi:

- Id_utente : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Username : un campo opzionale fornito dagli utenti al momento della registrazione e viene utilizzato durante la fase di login
- Pasw : campo obbligatorio in cui viene salvata la password che permetterà all’utente di autenticarsi, per ragioni di sicurezza è salvata in MD5 (Message Digest algorithm 5) all’interno del database
- Email : campo obbligatorio, insieme al campo “pasw” compone le credenziali di login utilizzate dagli utenti per autenticarsi
- Nome
- Cognome
- Sesso
- Indirizzo
- Telefono1 : numero di telefono principale
- Telefono2 : numero di telefono secondario

- Data_nascita : per facilitare la gestione delle date evitando problemi di localizzazione e per ragioni di ottimizzazione dello spazio tutte le date nel database sono in formato numerico intero (tempo Unix) rappresentato come offset in secondi rispetto alla mezzanotte (UTC) del 1° gennaio 1970 (detta epoca).
- Città
- Provincia : sigla della provincia
- Data_reg : rappresenta la data di registrazione, come la data di nascita è salvata in formato numerico e viene generata automaticamente al momento della registrazione

La tabella utenti non riesce ad esprimere alcuni concetti fondamentali legati all'utente registrato, come ad esempio il suo livello di privilegi o la classe di appartenenza, per ovviare a queste problematiche e avendo un occhio di riguardo alla complessità del database si è scelto di salvare tutte queste informazioni all'interno della tabella "utenti_meta".

Tale tabella sostituisce di fatto alcune tabelle aggiuntive che potevano essere create come ad esempio materie, classi, e le tabelle associative per questi dati. Le informazioni all'interno della tabella sono organizzate nel seguente modo:

- Id_meta : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Id_utente : id dell'utente a cui è riferito il record della tabella
- Key_name : stringa rappresentante un chiave generica per il dato da registrare. In questo modo la tabella è estremamente flessibile anche a future aggiunte di dati specifici per l'utente.
- Key_value : valore del campo associato alla chiave

Esaminando in modo più approfondito la tabella possiamo notare la flessibilità che questo tipo di struttura comporta, per chiarire facciamo qualche esempio utilizzando le chiavi fondamentali dell'applicazione :

- "user_lvl" : identifica il livello di privilegi dell'utente. Gli studenti hanno il numero più alto (99),decrementando tale numero si arriva al livello di tutti gli altri utenti, i genitori (3) , gli insegnanti (2) , il personale della segreteria (1) e gli amministratori (0). La scelta di avere i minori privilegi col numero più alto crea un ulteriore flessibilità qualora si ritenesse opportuno creare dei livelli di utenza intermedi.
- "Learn_in" : questa chiave indica la classe di appartenenza relativa ai soli studenti, il valore associato è la sigla della classe ad esempio "5A".

- “Teach” : sicuramente la chiave più complessa, racchiude le informazioni relative agli insegnanti. Il valore di questa chiave è un array che ha per chiavi il nome delle classi (1C,2B,4E ecc..) e per valore un ulteriore array rappresentante le materie insegnate nelle relative classi. La scelta di avere un array per le classi deriva sempre da un concetto di massima flessibilità e adattabilità al maggior numero possibile di casi che si possono presentare nei diversi ambienti in cui l’applicazione può essere inserita. L’array è memorizzato all’interno del database come una stringa (serializzazione).

Osservando la tabella da analizzare “genitori_rel” si nota che è composta dai campi :

- Id_rel : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Id_genitore : id utente del genitore
- Id_studente : id utente del figlio

Banalmente la tabella rappresenta le associazioni tra genitori e figli.

La prossima tabella da analizzare è “voti” formata dai campi :

- Id_voto : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Voto : il voto è salvato come stringa così da permettere anche pseudo-voti (es: 8+ o 7/8)
- Note : eventuali note associate al voto, rappresenta principalmente la natura del voto stesso (es: compito in classe, interrogazione orale, compito di recupero, ecc...)
- Id_studente : id utente dello studente a cui è associato il voto
- Id_insegnante : id utente dell’insegnante che ha registrato il voto
- Materia
- Data

Una delle operazioni principali di cui l’applicazione si occupa è l’appello, come supporto a tale funzione era necessario creare una tabella che memorizzasse le assenze, presenze e ritardi nel corso del tempo, per questo esiste la tabella “utenti_presenze”. I campi della tabella sono :

- Id_presenza : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Inserting_user : id utente di colui che registra la presenza, il nome del campo è generico e non ad esempio “id_insegnante” per un eventuale sistema di presenze riguardante gli insegnanti registrate dalla segreteria
- Target_user : id utente a cui è associata la presenza, assenza o ritardo, anche in questo caso il nome del campo è generico per gli stessi motivi di “inserting_user”.

- Tipologia : assenza, presenza o ritardo
- Tempo_ritardo : nel caso in cui la tipologia sia “ritardo” memorizza la quantità del ritardo
- Note
- Giustificata : flag utilizzato per identificare se l’assenza o il ritardo sono stati puntualmente giustificati dallo studente
- Data_reg : data in cui viene registrato l’evento
- Data : data originale dell’evento

L’ultima tabella è “comunicazioni”, fornisce il supporto per le funzionalità di comunicazione tra utenti, i campi sono :

- Id_comunicazione : chiave primaria della tabella, rappresenta un numero intero auto incrementale univoco
- Testo
- Tipo : tipologia della comunicazione (es. avviso, comunicazione generica, circolare, ecc..)
- Id_mit : id utente del mittente
- Id_dest : id utente del destinatario
- Data

Le comunicazioni verranno generate dagli insegnanti o dal personale della segreteria tramite un’apposita view, qualora il destinatario fosse uno studente una copia della comunicazione viene automaticamente inviata dal sistema anche al genitore.

Questo sistema di comunicazioni unito alla registrazione dei voti fornisce la base del controllo che i genitori avranno a disposizione sui propri figli.

4.4 Analisi dell'interfaccia utente

Le specifiche iniziali per l'interfaccia utente richiedevano un'applicazione flessibile, veloce ed intuitiva predisposta per funzionare su smartphone e tablet.

L'idea è di avere un menù laterale dal quale accedere alle funzionalità più frequentemente utilizzate ed uno spazio centrale in cui far convergere il contenuto di ogni view.

Tale struttura richiede in Android l'utilizzo di Fragment disponibili solo dall'API 16 in avanti (versione 4.1.x)

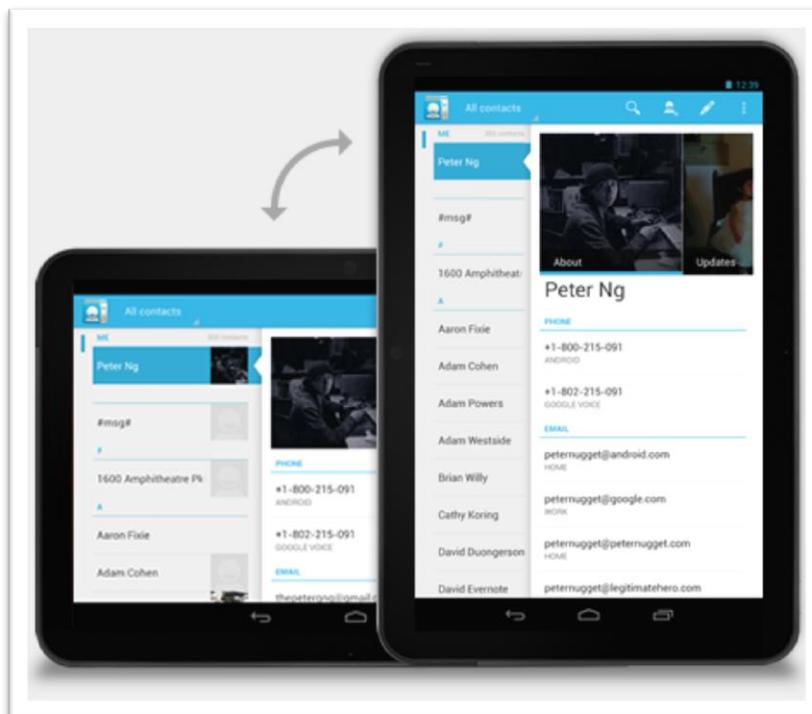


Illustrazione 7 : Bozza di interfaccia utente

La prima view presentata all'utente sarà un login dal quale l'utente inserendo il proprio nome utente e password (previa registrazione) sarà abilitato all'utilizzo dell'applicazione.

Una volta autenticato l'utente avrà a disposizione un menù simile a quello presentato nell'illustrazione 7 dal quale accedere a tutte le funzionalità messe a disposizione.

Il menù sarà diverso per ogni utente adattandosi al livello di privilegio posseduto dall'utilizzatore loggato.

L'elenco delle funzionalità presenti nel menù laterale e associate ad ogni utente è il seguente:

- Amministratori:
 - Home
 - Amministrazione : accesso al pannello di amministrazione
 - Classi : accesso all'elenco delle classi
 - Logout
- Personale segreteria:
 - Home
 - Classi
 - Studenti : accesso all'elenco degli studenti
 - Comunicazioni : accesso al sistema di invio comunicazioni
 - Logout
- Docenti:
 - Home
 - Eventi recenti : elenco delle ultime attività svolte (voti inseriti,comunicazioni inviate)
 - Profilo : accesso al pannello di modifica dei dati relativi al proprio utente
 - Logout
- Genitori:
 - Home
 - I miei figli : elenco delle attività dei propri figli
 - Profilo
 - Logout
- Studenti:
 - Home
 - Eventi recenti
 - Profilo
 - Logout

Per ogni utente la vista iniziale (home) è diversa e presenta le informazioni più utili per l'utente loggato, agli studenti verrà presentato l'elenco degli ultimi voti ricevuti, ai docenti l'elenco delle classi per facilitare l'appello, alla segreteria l'elenco delle ultime comunicazioni inviate.

5. Implementazione

In questo capitolo viene trattato nello specifico il codice che compone il sistema nel suo complesso. La prima view presentata all'utente è il login e, come detto in precedenza, l'applicazione può essere utilizzata solo previa autenticazione. Il login si effettua tramite un nome utente o una email e una password e per registrarsi non bisogna far altro che avviare la procedura tramite l'apposito pulsante.

5.1 Login: la prima view di SchoolAssistant

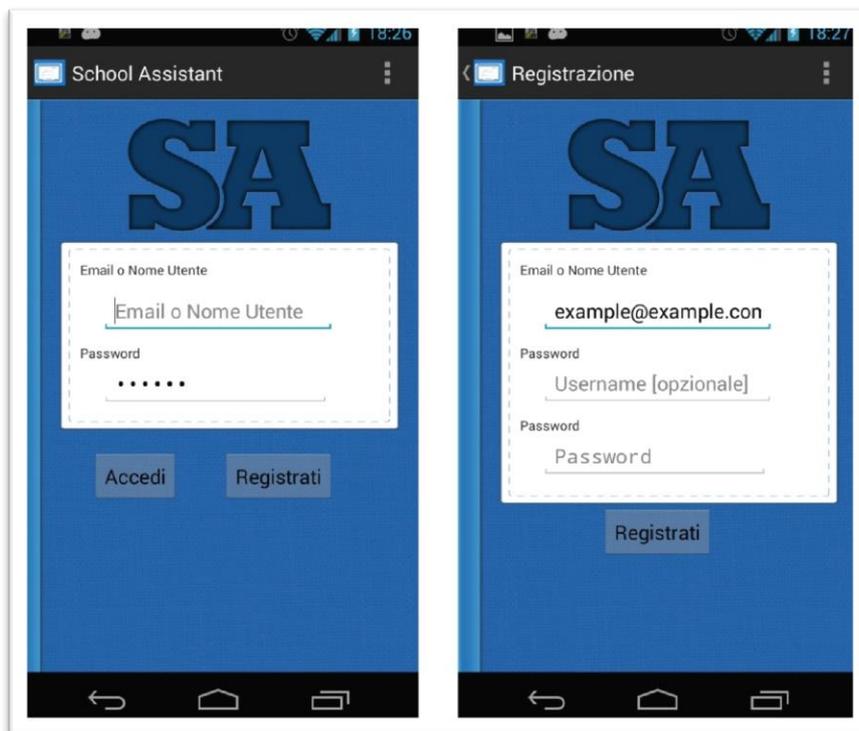


Illustrazione 8: login & registration view

Nel caso in cui le credenziali inserite siano corrette l'applicazione salva nelle *shared preference* i dati dell'utente così da renderli disponibili per tutta la sessione di utilizzo. Tutte le richieste al server effettuate dall'applicazione sono eseguite da task asincroni che permettono di continuare ad utilizzare l'applicazione durante l'elaborazione da parte del server.

```

public class UserLoginTask extends AsyncTask<Void, Void, Boolean> {
    @Override
    protected Boolean doInBackground(Void... params) {
        JSONObject jsonObj = null;
        Integer Result = 0;
        jsonObj = new JSONObject(ResponseData);
        Result = Integer.parseInt(jsonObj.getString("result"));
        if( Result == 1){
            HashMap<String,String> newuser = new HashMap<String, String>();
newuser.put(SessionManagement.KEY_USERID,jsonObj.getString("logged_user_id"));
            newuser.put(SessionManagement.KEY_USERNAME,
jsonObj.getString("logged_user_username"));
            newuser.put(SessionManagement.KEY_NAME,
jsonObj.getString("logged_user_nome"));
            newuser.put(SessionManagement.KEY_LASTNAME,
jsonObj.getString("logged_user_cognome"));
            newuser.put(SessionManagement.KEY_EMAIL,
jsonObj.getString("logged_user_email"));
            /* [...] Altri dati utente*/
            UtenteLoggato utente = new UtenteLoggato(newuser);
            session.createLoginSession(utente);
            return true;
        }
        return false;
    }

    @Override
    protected void onPostExecute(final Boolean success) {
        mAuthTask = null;
        showProgress(false);
        loginResult = success;

        if (!success){
            mPasswordView
                .setError(getString(R.string.error_incorrect_password));
            mPasswordView.requestFocus();
            Toast.makeText(getApplicationContext(),
R.string.error_incorrect_password, Toast.LENGTH_LONG).show();
        }else{
            Intent i = new Intent(_context, MainActivity.class);
            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            _context.startActivity(i);
            finish();
        }
    }

    @Override
    protected void onCancelled() {
        mAuthTask = null;
        showProgress(false);
    }
}

```

5.2 Home view

La “home view” è la view principale dell’applicazione, cambia a seconda del livello dell’utente autenticato presentandogli da subito le funzionalità peculiari. Anche il menù laterale si adatta a questi cambiamenti così da permettere all’utente di accedere solo alle funzioni a cui è abilitato.



Illustrazione 9: Home view Docenti

In fase di analisi si è deciso di modificare la Home View come segue:

- Ai docenti si presenta la lista delle classi di cui sono responsabili, cliccando su una classe è possibile visualizzare l’elenco degli studenti, registrare le presenze e accedere alle funzionalità disponibili per ogni singolo studente
- Agli studenti viene presentata la lista degli ultimi eventi che li riguardano composta dall’elenco dei voti e delle comunicazioni ricevute
- Agli amministratori e al personale della segreteria si presenta un elenco delle comunicazioni inviate e ricevute
- Ai genitori si presenta un elenco di voti e comunicazioni riguardanti i propri figli

5.3 Il menù laterale

Il menù laterale, sempre accessibile dall'angolo in alto a sinistra, permette agli utenti di avere sempre a portata di mano le funzioni principali, in generale le voci del menù sono le seguenti:

- Home
- Amministrazione
- Classi
- Eventi recenti
- Comunicazioni
- I miei figli
- Profilo
- Logout

Come detto in precedenza ogni utente avrà a disposizione solo un sottoinsieme di tutte le funzionalità presenti.

5.4 Elenco studenti e pannello azioni

L'elenco studenti è una delle view più complesse dal punto di vista dello sviluppo, da qui si possono registrare le presenze per la giornata in corso oppure consultare quelle dei giorni passati selezionando la data utilizzando l'apposito calendario.

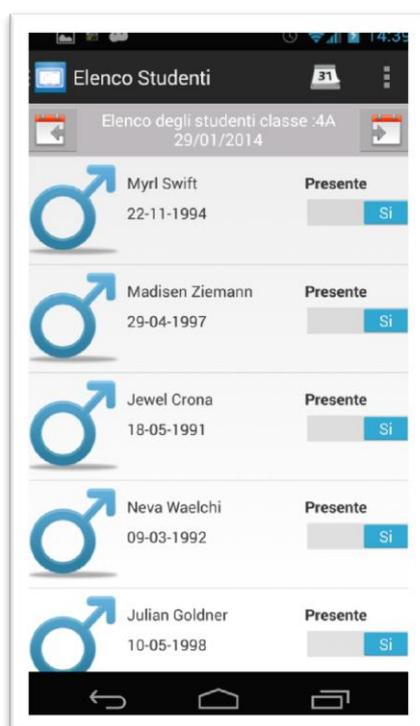


Illustrazione 10: View elenco studenti

Si è deciso di affiancare ad ogni studente uno *switch* da cui registrare facilmente l'assenza o la presenza dello studente. Di default tutti gli studenti vengono registrati come presenti non appena si accede a questa view, sarà poi compito del docente segnalare gli assenti tramite l'apposito switch.

Nel caso fosse necessario segnalare un ritardo o aggiungere delle note per descrivere un'assenza è sufficiente cliccare sullo studente selezionato per accedere al pannello delle azioni. Da questo pannello è inoltre possibile assegnare un voto, inviare delle segnalazioni o visualizzare la lista dei voti dello studente.

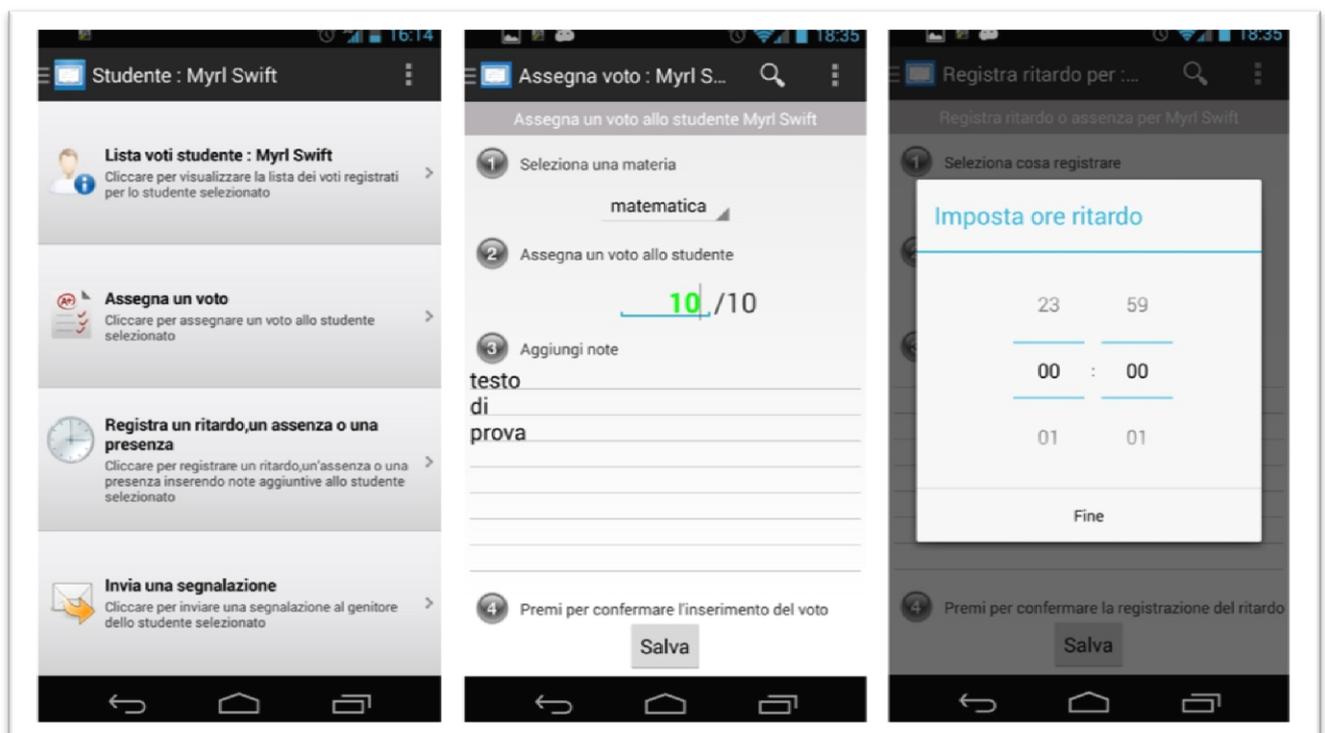


Illustrazione 11 : Pannello azioni docente

Ogni view è predisposta con un elenco puntato che guida il docente nel compiere l'azione richiesta, si è cercato di mantenere la grafica il più uniforme possibile così da non confondere l'utente.

Premendo il pulsante salva i dati vengono trasmessi al server che si occupa della registrazione all'interno del database; il risultato dell'operazione viene visualizzato in un messaggio *toast* che permette all'utente in caso di errore di venire a conoscenza dei problemi riscontrati.

La lista dei voti comprende di default tutte le discipline, è tuttavia possibile tramite il pulsante “cerca” filtrare i voti in base alla materia selezionata. Nella stessa view viene data la possibilità di visualizzare la media voti totale o filtrata per materia.

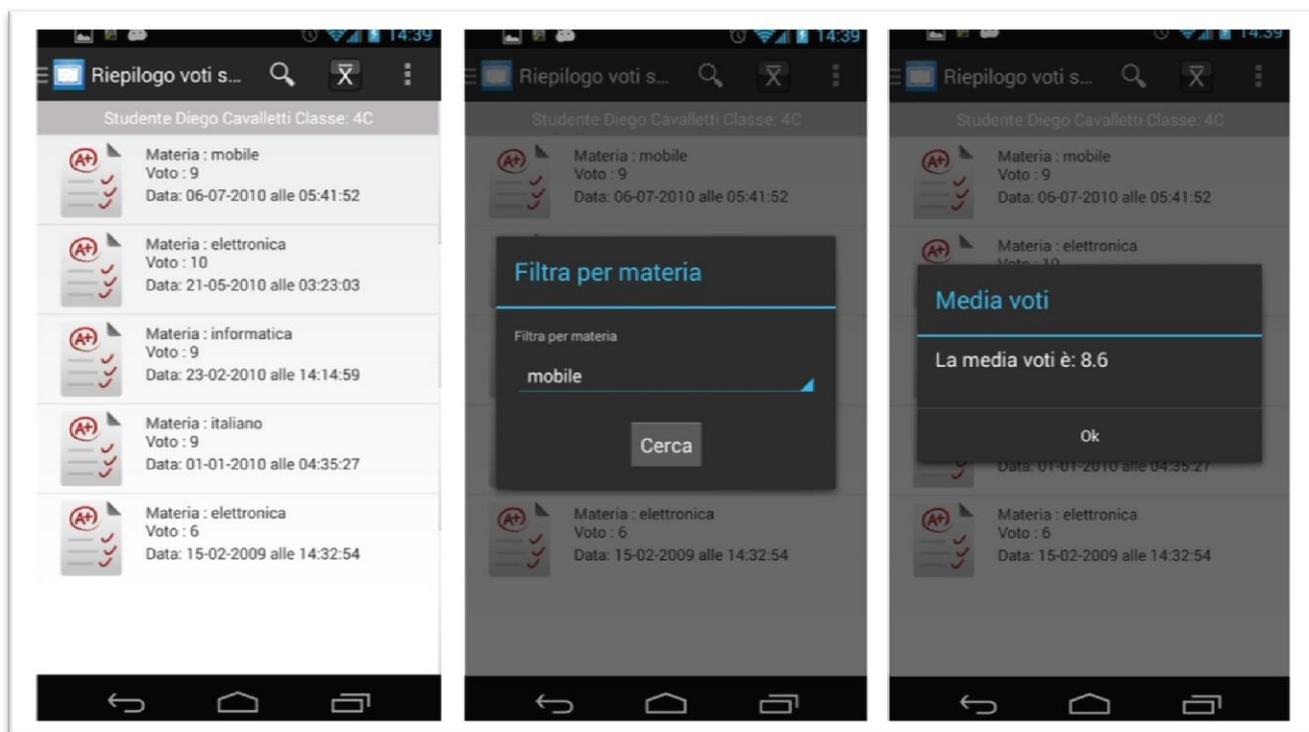


Illustrazione 12: Lista voti

Una volta filtrati i voti per materia è possibile ritornare alla lista completa semplicemente premendo il pulsante “indietro” del proprio smartphone.

L’elenco delle materie su cui filtrare viene dinamicamente modificato a seconda dei voti registrati per l’alunno selezionato, sono infatti presenti come filtri solo le discipline in cui lo studente ha ricevuto almeno un voto.

La maggior parte delle funzionalità legate ai docenti fin qui illustrate sono gestite dalla classe *teacher* che estende la classe “user”. La creazione di classi indipendenti aumenta la riusabilità e la scalabilità del codice rendendolo più flessibile ai nuovi aggiornamenti.

5.5 Eventi recenti

La view degli eventi recenti viene utilizzata, opportunamente filtrata, per tutti gli utenti.

In questa view viene presentato un elenco di voti e/o comunicazioni che interessano l'utente attualmente autenticato, cliccando sulle righe di questo elenco è possibile accedere alla view dei dettagli associata al tipo di evento selezionato.

Ad ogni tipologia di evento, sia esso una comunicazione in arrivo, una in uscita o un voto è assegnata un'icona differente per facilitarne il riconoscimento da parte dell'utente.

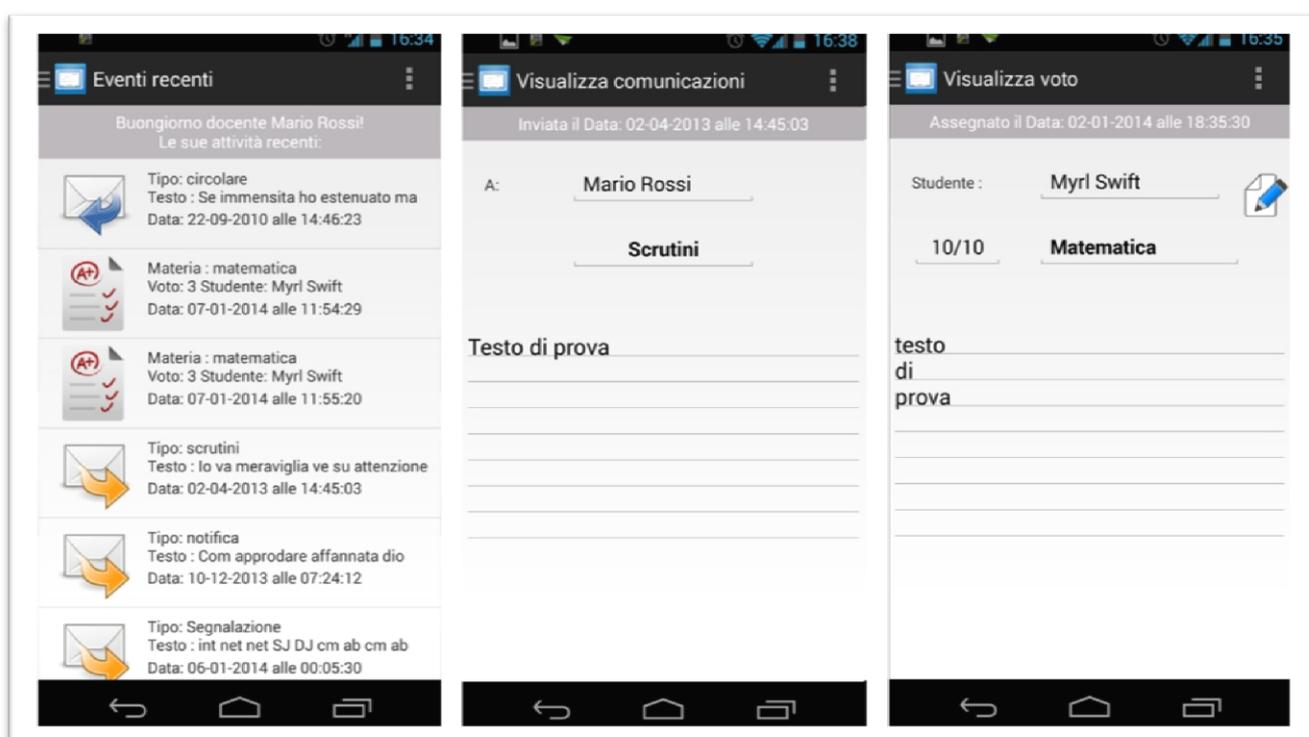


Illustrazione 13: Lista eventi e dettagli

Se l'utente correntemente autenticato dispone di abbastanza privilegi, come si può notare dall'illustrazione 13, è presente un pulsante di modifica nella *view* di dettaglio del voto. Questa funzione è nascosta se l'utente non ha i privilegi necessari.

5.6 Amministrazione

La parte amministrativa dell'applicazione consente agli utenti di livello "admin" di eseguire le operazioni indispensabili per l'inizializzazione dell'applicazione quali aggiunta di strutture, classi e utenti. Gli amministratori sono gli unici utenti a cui è permesso accedere a queste operazioni.

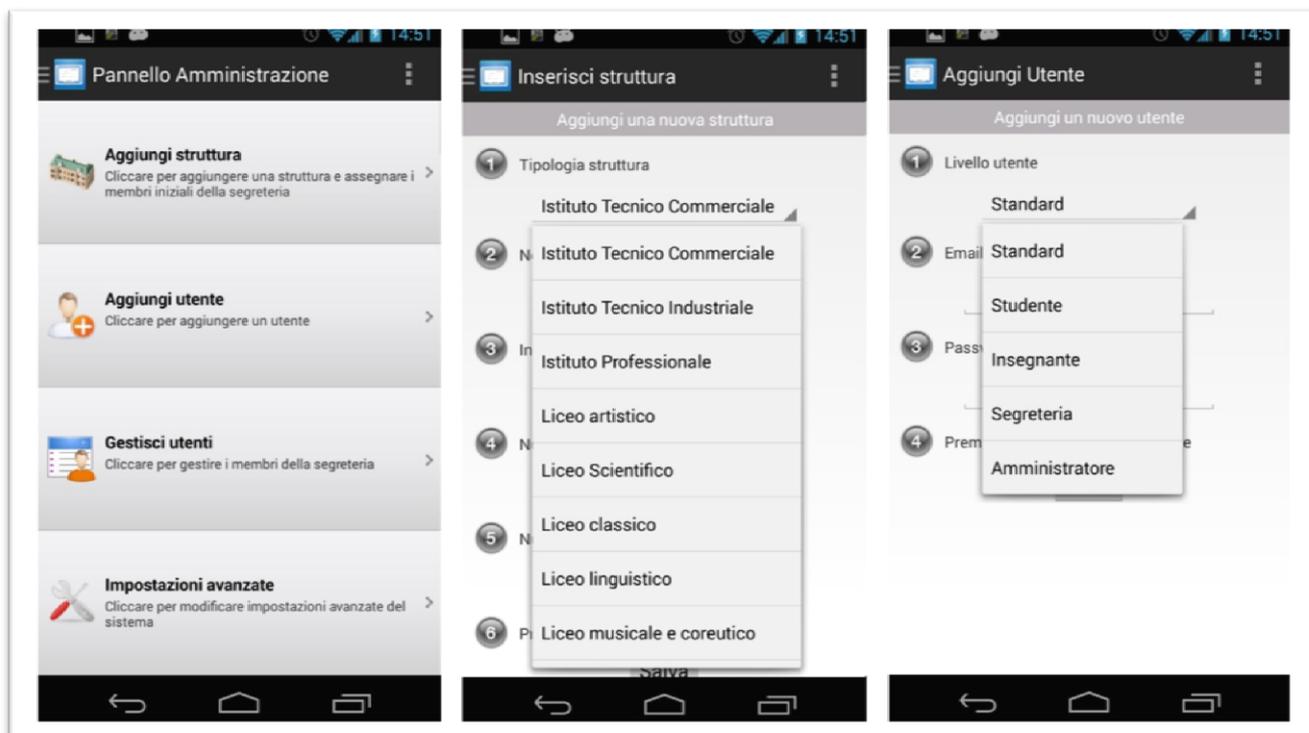


Illustrazione 14: Amministrazione

Tutte le operazioni effettuabili dagli amministratori sono gestite da una classe di supporto dal nome "admin.java". La classe *admin* come la classe *teacher* estende la classe "user" dalla quale eredita i metodi di basi necessari al corretto funzionamento.

Questo tipo di struttura permette di aggiungere facilmente nuove funzionalità all'applicazione.

L'ultima operazione di rilievo eseguibile dai soli amministratori è l'aggiunta di nuove classi, tramite l'apposita view vengono mostrate tutte le classi al momento presenti, con la possibilità di aggiungerne di nuove semplicemente cliccando l'apposito pulsante in alto a destra e inserendo un nome per la nuova classe.



Illustrazione 15 : Aggiunta di una nuova classe

L'assegnazione degli utenti ad una determinata classe può avvenire durante la fase di creazione dello stesso oppure dall'opportuno pannello di modifica selezionando un utente dall'elenco. Se si sceglie di visualizzare l'elenco di tutti gli utenti può essere difficile trovare l'interessato, per questo è stata implementata una funzione di filtro in base al livello di utente così da facilitare e velocizzare la ricerca.

5.7 Scambio di dati col server

Lo scambio di dati tra l'applicazione e il server avviene tramite richieste asincrone inviate dall'applicazione alle quali il server risponde con dati in formato JSON come descritto durante l'analisi.

Prendiamo ad esempio la gestione degli eventi recenti comune alla maggior parte degli utenti, al server viene inviata una richiesta contenente l'id dell'utente autenticato.

La richiesta viene gestita da una classe appositamente creata, il ConnectionManager ed è formata nel modo seguente :

```
public String RecentEventsRequest(String user_id) throws InterruptedException
{
    String data = "";
    try{
        data = URLEncoder.encode("user_id", "UTF-8") + "=" +
URLEncoder.encode(user_id, "UTF-8");

        if(my_chilids){
            data += "&"+URLEncoder.encode("action", "UTF-8") + "=" +
URLEncoder.encode("childEvents", "UTF-8");
        }else{
            data += "&"+URLEncoder.encode("action", "UTF-8") + "=" +
URLEncoder.encode("events", "UTF-8");
        }
    }catch(UnsupportedEncodingException ex){
        //encoding non supportato
    }
    Log.i("Result from recent_events.php",data);
    ConnectionManager cm = ConnectionManager.getInstance();
    String requestResult = cm.CreateRequest(data, "recent_events.php");
    return requestResult;
}
```

All'interno della variabile "data" viene costruita la stringa contenente i dati da inviare al server, in questo caso è piuttosto semplice in quanto formata soltanto da due campi: "user_id" ed "action". User id è ovviamente l'id dell'utente mentre action è una variabile che consente al server di capire quale azione eseguire con i dati ricevuti, nel caso generale vale "events" ("childEvents" è solo nel caso in cui si debba costruire la lista degli eventi per un genitore).

La variabile "data" insieme al nome della pagina PHP lato server che si occupa della gestione degli eventi recenti viene passata al ConnectionManager che si occupa di inviare la richiesta HTTP al server.

L'invio dei dati al server avviene all'interno della funzione "doInBackground" del task asincrono così da non bloccare l'esecuzione dell'applicazione durante l'attesa della risposta da parte del server. Una volta ricevuta la risposta all'interno della stessa funzione ci si occupa del cosiddetto "parse" ossia la lettura del risultato in formato JSON.

```
String responseData = "";
JSONObject jsonObj = null;
Integer result = 0;
responseData = RecentEventsRequest(user_id);
jsonObj = new JSONObject(responseData);
result = Integer.parseInt(jsonObj.getString("result"));
if( result == 1){
    JSONObject com_set = jsonObj.getJSONObject("com_set");

    for(Iterator<String> iter = com_set.keys();iter.hasNext();) {
        String key = iter.next();
        JSONArray evt = null;
        if(key.equals("voti")){
            //Parse Json voti
        }else if(key.equals(COMUNICAZIONE_ARRIVO) ||
key.equals(COMUNICAZIONE_USCITA)){
            //Parse Json comunicazioni
        }
    }
    return true;
}
SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy
HH:mm:ss",Locale.getDefault());
Date today = new Date();
payload.descriptions.add("Non ci sono eventi recenti");
payload.date.add(dateFormat.format(today));
payload.itemsId.add(-1);
payload.imageId.add(R.drawable.none);
payload.type.add("none");
payload.sub_type.add("none");
return true;
```

La prima cosa che si può notare è che si va a controllare un parametro della risposta denominato "result", tale parametro informa l'applicazione sull'esito dell'operazione.

Se l'operazione è andata a buon fine per convenzione (valida all'interno di tutta l'applicazione) il parametro "result" ha valore 1, si può quindi procedere al parse dei restanti dati contenuti nella risposta.

Durante il parse degli eventi recenti si possono incontrare due tipologie di oggetti:

- Voti
- Comunicazioni divise in
 - Comunicazioni inviate
 - Comunicazioni ricevute

Esaminando un oggetto di tipo voto si può vedere da quali campi è formato e come l'applicazione li estrae dal JSON ricevuto salvandoli poi in un oggetto di classe "RecentActivityDataPayload" chiamato "payload".

```
evt = com_set.getJSONArray("voti");
int len = evt.length();
for (int i = 0; i < len; i++) {
    JSONObject arr_voto= evt.getJSONObject(i);
    String mat = arr_voto.getString("materia");
    String voto = arr_voto.getString("voto");
    String data_f = arr_voto.getString("data_f");
    String time = arr_voto.getString("time");
    String nome_studente = arr_voto.getString("nome_studente");
    String nome_insegnante = arr_voto.getString("nome_insegnante");
    payload.sub_type.add(mat);
    if(user_lvl == MainActivity.TEACHER_LVL)
        payload.descriptions.add("Materia : "+mat+" \nVoto: "+voto+" Studente:
"+nome_studente);
    else
        payload.descriptions.add("Materia : "+mat+" \nVoto: "+voto+" Insegnante:
"+nome_insegnante);
    payload.date.add("Data: "+data_f+" alle "+time);
    payload.imageId.add(R.drawable.marks_big);
    payload.itemsId.add(arr_voto.getInt("id_voto"));
    payload.type.add(TIPO_VOTO);
}
```

Il server raggruppa il risultato costruendo tre array, uno per ogni tipo di oggetto possibile. Ogni array è a sua volta composto da dei sub-array che rappresentano il singolo oggetto di ciascun tipo.

Se invece durante il parse l'applicazione incontra un oggetto di tipo comunicazione, sia essa inviata o ricevuta, è necessario eseguire altre operazioni così da gestire in modo appropriato i campi che formano l'oggetto comunicazione.

```
if( key.equals(COMUNICAZIONE_ARRIVO) ){
    evt = com_set.getJSONArray(COMUNICAZIONE_ARRIVO);
}else if( key.equals(COMUNICAZIONE_USCITA) ){
    evt = com_set.getJSONArray(COMUNICAZIONE_USCITA);
}
int len = evt.length();
for (int i = 0; i < len; i++) {
    if( key.equals(COMUNICAZIONE_ARRIVO) ){
        payload.imageId.add(R.drawable.mail_in_big);
    }else if( key.equals(COMUNICAZIONE_USCITA) ){
        payload.imageId.add(R.drawable.mail_out_big);
    }
    JSONObject arr_com= evt.getJSONObject(i);
    String testo = arr_com.getString("testo");
    String tipo = arr_com.getString("tipo");
    String data_f = arr_com.getString("data_f");
    String time = arr_com.getString("time");
    payload.sub_type.add(tipo);
    payload.descriptions.add("Tipo: "+tipo+"\nTesto : "+testo);
    payload.date.add("Data: "+data_f+" alle "+time);
    payload.itemsId.add(arr_com.getInt("id_comunicazione"));
    payload.type.add(TIPO_COMUNICAZIONE);
}
```

Come nel caso dei voti anche le comunicazioni sono raggruppate all'interno della propria macro categoria suddivise in entranti ed uscenti.

6. Sviluppi futuri

L'applicazione oggetto di questa tesi ricopre un set base delle funzionalità associate ai registri elettronici, è quindi un punto di partenza per portare le potenzialità di uno strumento al momento accessibile solo via web negli smartphone delle persone nell'ambito scolastico.

La comodità dell'applicazione sta nel fornire agli utenti informazioni in tempo reale e facilmente accessibili, con la possibilità di inviare notifiche visualizzabili direttamente dal proprio dispositivo anche senza accedere direttamente al servizio.

Tra le funzionalità da sviluppare in futuro si trovano sicuramente:

- Ottimizzazione del codice
- Calendario lezioni
- Ottimizzazione dello scambio dati client-server
- Notifiche in tempo reale

7. Conclusioni

SchoolAssistant ha l'obiettivo di integrare i servizi offerti dagli edifici scolastici ai propri iscritti, siano essi docenti, studenti, genitori o personale della segreteria affiancando il registro elettronico accessibile via web.

Le potenzialità di questa applicazione sono sicuramente vastissime e potrebbero comprendere o arricchire altri servizi, in un mondo in cui il tempo è un bene sempre più prezioso la possibilità di controllare i propri figli direttamente dal proprio smartphone è una prospettiva allettante.

8. Bibliografia

- <http://www.fastweb.it/smartphone-e-gadget/la-storia-di-android/>
- <http://news.softonic.it/news-2013-sistemi-operativi-mobile>
- <http://www.iphoneitalia.com/da-simon-alliphone-la-storia-degli-smartphone-273474.html>
- <http://www.studenti.it/superiori/scuola/registro-elettronico-come-funziona-foto-video-scuole.php>
- <http://www.itespresso.it/idc-il-mercato-smartphone-record-88904.html>
- <http://developer.android.com/about/dashboards/index.html>
- <http://www.ridble.com/storia-apple-ios/>

9. Indice

| | |
|---|----|
| Introduzione | 3 |
| 1. Storia degli smartphone | 5 |
| 1.1. Simon : il primo “smartphone” | 6 |
| 1.2. Palm Pilot : il precursore dell’avvento degli smartphone | 6 |
| 1.3. Nokia 9110 Communicator : l’esempio da seguire | 6 |
| 1.4. BlackBerry 5810 : Il vero smartphone | 7 |
| 1.5. Apple Iphone 1 : lo smartphone per tutti | 8 |
| 1.6. Gli smartphone al giorno d’oggi | 9 |
| 2. Il cuore degli smartphone : Il sistema operativo | 11 |
| 2.1. Android : il SO mobile più diffuso | 12 |
| 2.2. Windows Phone : un SO in crescita | 13 |
| 2.3. Symbian OS : un progetto abbandonato | 14 |
| 2.4. Blackberry | 15 |
| 2.5. iOS | 16 |
| 3. La tecnologia nell’istruzione | 17 |
| 4. Progettazione | 19 |
| 4.1. Analisi dell’ambiente applicativo e degli utenti finali | 19 |
| 4.2. Analisi delle funzionalità richieste | 19 |
| 4.2.1. Amministratore di sistema | 20 |
| 4.2.2. Personale della segreteria | 20 |
| 4.2.3. Docenti | 21 |
| 4.2.4. Genitori | 21 |
| 4.2.5. Studenti | 21 |
| 4.3. Analisi strutturale dell’applicazione | 22 |
| 4.3.1. La scelta del sistema operativo | 22 |
| 4.3.2. Struttura delle comunicazioni | 25 |
| 4.3.3. Struttura del database | 25 |
| 4.4. Analisi dell’interfaccia utente | 31 |
| 5. Implementazione | 33 |
| 5.1. Login: la prima view di SchoolAssistant | 33 |
| 5.2. Home view | 35 |
| 5.3. Il menù laterale | 36 |
| 5.4. Elenco studenti e pannello azioni | 36 |
| 5.5. Eventi recenti | 39 |
| 5.6. Amministrazione | 40 |
| 5.7. Scambio di dati col server | 42 |
| 6. Sviluppi futuri | 47 |
| 7. Conclusioni | 49 |
| Bibliografia | 51 |

