VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION TECHNOLOGIES

Tadas Pocius

MOBILE AIRLINE TO PASSENGER COMMUNICATION
MOBILIOJI AVIAKOMPANIJŲ IR JŲ KELEIVIŲ KOMUNIKACIJA

Final master’s dissertation

Information Technologies Management study programme, state code 621I13001

Physical Sciences, Informatics

Vilnius, 2014

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

TVIRTINU

Katedros vedėjas

(Parašas)

(Vardas, pavardė)

(Data)

Tadas Pocius

MOBILE AIRLINE TO PASSENGER COMMUNICATION
MOBILIOJI AVIAKOMPANIJŲ IR JŲ KELEIVIŲ KOMUNIKACIJA

Baigiamasis magistro darbas

Informacinių Technologijų Valdymo studijų programa, valstybinis kodas 621I13001

Informatikos mokslų studijų kryptis

Vadovas	(Moksl. laipsnis/pedag. vardas, vardas, pavardė)	(Parašas)	(Data)
Lietuvių kalbos konsultantas	(Moksl. laipsnis/pedag. vardas, vardas, pavardė)	(Parašas)	(Data)

Vilnius, 2014

(the document of Declaration of Authorship in the Final Degree Paper)

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Tadas Pocius, 20073854

(Student's given name, family name, certificate number)

Faculty of Fundamental Sciences

(Faculty)

Information Technology Management, ITVmit-12

(Study programme, academic group no.)

DECLARATION OF AUTHORSHIP
IN THE FINAL DEGREE PAPER

June 1, 2014

I declare that my Final Degree Paper entitled „Mobile Airline to Passenger Communication" is entirely my own work. The title was confirmed on December 9, 2013 by Faculty Dean's order No. 400fm. I have clearly signalled the presence of quoted or paraphrased material and referenced all sources.

The academic supervisor of my Final Degree Paper is Prof Dr Habil Genadijus Kulvietis.

No contribution of any other person was obtained, nor did I buy my Final Degree Paper.

[illegible]

(Signature)

Tadas Pocius

(Given name, family name)

Vilniaus Gedimino technikos universitetas **Fundamentinių Mokslų** fakultetas **Informacinių Technologijų** katedra	ISBN ISSN Egz. sk. ……….. Data ….-….-….

Informacinių Technologijų Valdymo studijų programos baigiamasis magistro darbas

Pavadinimas **Mobilioji aviakompanijų ir jų keleivių komunikacija**

Autorius **Tadas Pocius** Vadovas prof. Habil. dr. **Genadijus Kulvietis**

Kalba

	lietuvių
X	užsienio

Anotacija

Baigiamajame magistro darbe atliekama kompanijos „15below“ naudojamos avialinijų komunikacijos su keleiviais platformos „PASNGR“ analizė. Atskirai atliekama trumpa kiekvieno „PASNGR“ modulio apžvalga įvertinant taikymo galimybes. Be to, kadangi platforma yra priklausoma nuo globaliųjų paskirstymo sistemų, apžvelgiamos ir jos apimant procesus, bei naudojamus duomenis.

Darbe yra numatomas sukurti mobiliosios aplikacijos prototipas surištas su „PASNGR“ platforma. Tam tikslui yra atliekama integracijos analizė, apžvelgiant komponentus, duomenis ir funkcijas. Galiausiai yra sukuriamas veiklos proceso modelis atvaizduojantis mobiliosios aplikacijos prototipo veikimo principus, bei interakciją su platforma „PASNGR“.

Darbo eigoje apžvelgiamos mobiliųjų aplikacijų technologijos, bei įrankiai joms kurti. Taip parenkamas priimtiniausias spendimas esamai kompanijos „15below“ programuotojų aplinkai ir lemiantis minimalias sąnaudas.

Galiausiai sukuriamas mobiliosios aplikacijos prototipas veikiantis Android ir iOS platformose, gebantis atsisiųsti elektroninius bilietus į mobilųjį įrenginį ir jame išsaugoti naudojimui neprisijungus prie interneto, leidžiantis naudotojui skenuoti brūkšninį kodą oro uoste, po skrydžio pateikti nuomonę apie gautas avialinijų paslaugas, priimti pranešimus apie skrydžių pakeitimus, bei leisti juos patvirtinti.

Darbą sudaro 6 dalys: įvadas, esamos platformos analizė, rezervacijos sistemų analizė, mobiliosios aplikacijos integracijos analizė, mobiliosios aplikacijos prototipas, išvados ir siūlymai, literatūros sąrašas.

Darbo apimtis – 57 p. teksto be priedų, 42 iliustr., 1 lent., 14 bibliografinių šaltinių.

Atskirai pridedami darbo priedai.

Prasminiai žodžiai: GDS, rezervacijos sistemos, integracija, veiklos procesų modelis, mobiliosios platformos, Adroid, iOS, Adobe Phonegap.

Vilnius Gediminas Technical University
Faculty of **Fundamental Sciences**
Department of **Information Technologies**

ISBN ISSN
Copies No. ....
Date ....-....-....

Information Technologies Management study programme master thesis.

Title: **Mobile Airline to Passenger Communication**

Author **Tadas Pocius** Academic supervisor **Prof. Dr. habil. Genadijus Kulvietis**

Thesis language

[ ] Lithuanian
[✓ X] Foreign (English)

Annotation

In the final master's dissertation analysis on current 15below communication platform PASNGR. In addition to this there is a short analysis done on each of PASNGR components to find out usage possibilities. Furthermore, the platform is dependant on Reservation Systems, so analysis on them is done as well, including processes and data.

The goal of this dissertation is to create a mobile app integrated into PASNGR. For this reason integration analysis is done, by reviewing components, data and functions. After analysis is done, Business Process Model is created to represent the workflow of the app, including internal processes and interaction with PASNGR.

Furthermore a review of mobile application technologies is done, including common tools used in development. By doing this there is the best solution for current 15below developers' environment found. The solution would mean the least resources to be used to develop the app.

Finally a prototype of the mobile application is created, to show the possibilities. The app works on Android and iOS platforms, gives an ability for user to download itineraries and store for offline usage, an ability to scan a barcode at the airport straight from the app, give post flight experience feedback for the airline, receive schedule change notifications and accept them.

Structure: introduction, current platform analysis, reservation systems analysis, mobile application integration analysis, mobile application prototype, final results, references.

Thesis consist of: 57 p. text without appendixes, 42 pictures, 1 tables, 14 bibliographical entries.

Appendixes included.

Keywords: GDS, Reservation Systems, Integration, Business Process Model, Mobile Platforms, Adroid, iOS, Adobe Phonegap.

Table of contents

Glossary

API - Application Program Interface

APIS - Advanced Passenger Information System

APN - Apple Push Notification

B2B - Business to Business

CPT - Cross Platform Tool

CRM - Customer Relations Management

CRS - Computer Reservation System

CSS - Cascade Style Sheet

DCS - Distributed Control System

ESTA - Electronic System for Travel Authorization

FLIFO - Flight Info

GCM - Google Cloud Messaging

GDS - Global Distribution System

HTML - Hyper Text Mark-up Language

IATA - International Air Transport Association

IDE - Integrated Development Environment

IROP - Irregular operation

IT - Information Technologies

JSON - JavaScript Object Notation

MVC - Model View Control

PC - Personal Computer

PDF - Portable Document Format

PNR - Passenger Name Record

QML - Query Modeling Language

QR - Quick Response

REGID - Device Registration Id

RIA - Rich Internet Application

SMS - Short Messages System

SSR - Special Service Request

UI - User Interface

List of figures

List of tables

1. Introduction

Everyday millions of people are traveling on planes around the globe. For some of them, flight is not only a handy type of transport to reach remote locations, but also a stress cause. If a person feels nervous before a flight, just imagine how much of additional stress would mean when something goes wrong while on the ground just before the flight. For instance a flight ticket left at home. This scenario would mean long queuing at the airport check-in desk to get a new boarding pass, with the only one thought in mind of how not to miss the flight and get back home on time. Another situation which could lead to missed plane would be a schedule change. It is very important to let passenger know about alteration of his trip, so he could sort his trip without a rush.

15below ltd is a company who are global leaders in travel communications market. Their vision is simple – make journeys easier and more enjoyable by giving our clients and their passengers a stronger sense of control [1]. Company is providing a platform which uses Global Distribution Systems (GDS) to retrieve flights and passengers details and uses them mainly to send notifications like itineraries, booking confirmations, schedule change notifications, flight status notifications etc. 15below main sending channels are Email, Short Message System (SMS), Voice, Fax and Print. Company credo is "Calm. Connected.", but can you say "Connected" today, when mobile platforms are guiding the whole IT market and you don't have a competitive product for mobile devices?

Every time you want to download an application for your mobile devices you get a huge choice of options by the keyword you have entered. This gives an idea that the market of mobile applications could be stuffed with similar apps. So as a starting point it is necessary to make 15below developed platform analysis to see what services it can offer already, so we could later decide how to use these services to create a competitive app.

As I already mentioned 15below is retrieving details from GDS. GDS is like data cloud, so it is very important to do analysis on this data cloud as well, so we know what information we are dealing with.

Having a view of services already provided by 15below and the data is not enough to start working on mobile application. Especially when platform has no send channel dedicated for mobile devices, expect for SMS, which is more for cell phones than mobile devices. Mobile devices are not only smart phones, but tablets as well. So the next target would be to prepare a scenario of mobile application integration into already existing platform.

When analysis is done on all parts involved in the process, it is the time to use the theory in practice. In the other words the last step would be to create a concept of the mobile application integrated into current 15below communications platform, which could offer as much services provided by current platform as possible.

Finally the results must cover project’s expectations, to see what the effort would be to rollout this application to production.

2. Current platform analysis

2.1 INTRODUCTION

There are over 90000 commercial flights every day. This means there are millions of people traveling on planes every day. To get on board you need a ticket and it is a high chance that a ticket was generated and sent by 15below intelligent messaging platform called PASNGR. Retrieving flights and passengers details and itineraries sending is just one of many PASNGR functionalities. This target of this chapter is to review each of PASNGR modules and to make a decision which of them could be used in mobile devices. As the initial stage it is very important to decide whether any services of the current platform are possible in a mobile app.

2.2 SERVICES

15below business model is Business-to-Business (B2B), which means they are providing services for business. 15below customers are well known airlines and travel agencies (Figure 1). Customers are using the platform by their needs.

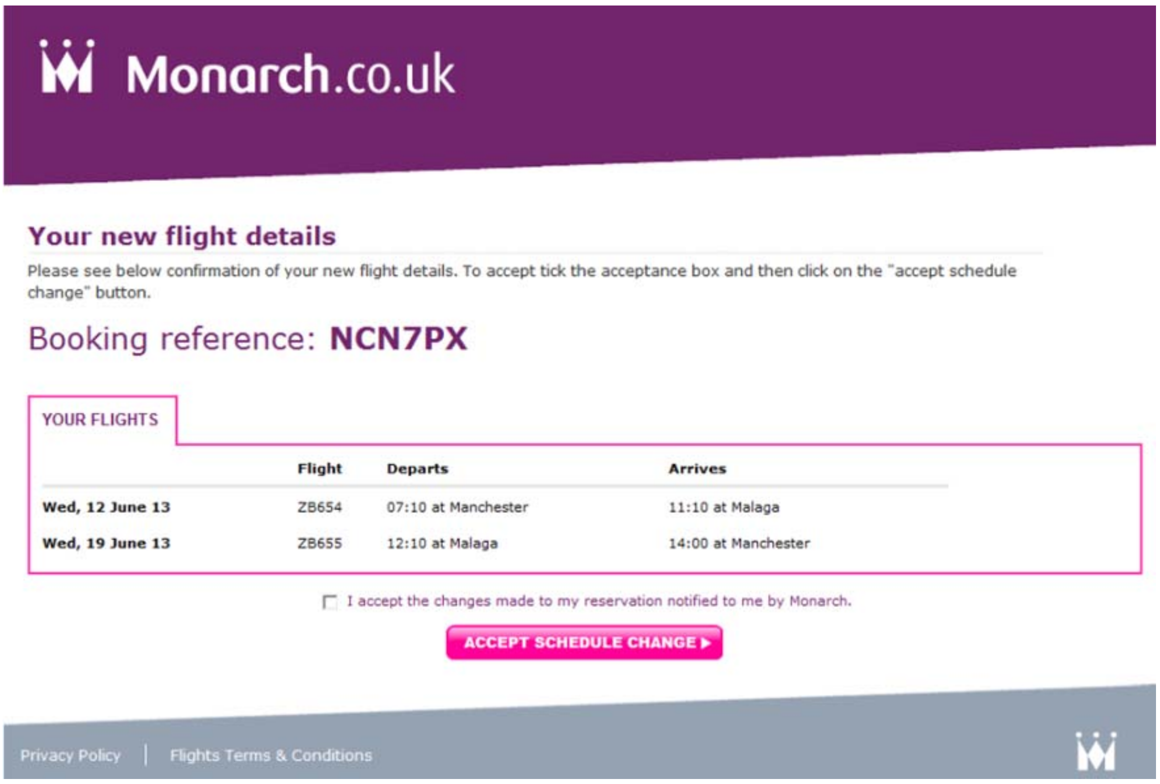
Every customer is choosing modules from a total of 9 available, by its needs. This is because every customer has its own business model and internal processes. PASNGR is the platform that could reduce cost of doing business, by replacing human resources and automate lots of day to day tasks.

So what are these modules?

Disruption

When you are about to take off, the last thing you want to happen is to get your flight cancelled or changed. However a storm, a volcano or just air traffic control strike can change your trip plans. In such cases communication is essential to avoid frustration and chaos taking control.
15below technology platform integrates directly with any reservations system. It keeps an airline on top of the situation by keeping its' customers informed and automating the extra tasks required to get everyone back on track.

Using branded message templates an airline can inform travelers about schedule changes, cancellations or ad hoc disruption quickly and easily. The customer can then chose to accept the proposed changes from their phone or computer, or opt for a credit or refund (Figure 1). These actions are automatically fed back into your reservations system so your staff can see exactly what is going on and focus their time contacting those they need to [2].



Your new flight details

Please see below confirmation of your new flight details. To accept tick the acceptance box and then click on the "accept schedule change" button.

Booking reference: **NCN7PX**

YOUR FLIGHTS

	Flight	Departs	Arrives
Wed, 12 June 13	ZB654	07:10 at Manchester	11:10 at Malaga
Wed, 19 June 13	ZB655	12:10 at Malaga	14:00 at Manchester

☐ I accept the changes made to my reservation notified to me by Monarch.

Figure 2. Schedule change acceptance website screenshot [1].

Disruption module key features:

- Handles refunds, declines, credits and claims
- Flexible schedule change and cancellation workflows
- Manages and schedules reminder notifications
- Automatically updates customer responses in an airline system
- Supports multiple channels (Email, SMS, Push, Voice)
- Integrates with CRMs and other systems

The benefit for an airline:

- Reduce inbound calls and avoid call centre throttling
- Increase operational capability and customer reach
- Improve inventory visibility
- Give service that builds customer trust and loyalty
- Reduce complaints

Flight and travel status

An airline can create smoother journeys by giving their passengers, staff and subscribers reliable flight information that's relevant to them. 15below Flight and Travel Status module integrates with airline FLIFO, reservations and other systems so they can send perfectly timed and accurate updates and reminders on boarding times, gate changes, cancellations or delays (Figure 2).

An airline can set up templates, schedules and automated workflows to keep everyone up to date with minimum fuss, or use the module to contact people manually if you need to. By linking in to their content management system or third party feeds airlines can also add personalized content or targeted advertising to their messages.

These kinds of notifications can improve everyone's experience - helping an airline to keep their ground operations running like clockwork, their passengers calm, and their subscribers ready at the arrival gates [2].

It is getting popular to hire drivers to meet up our guests or even us and we don't want to get lost while abroad due to flight change. Do we?

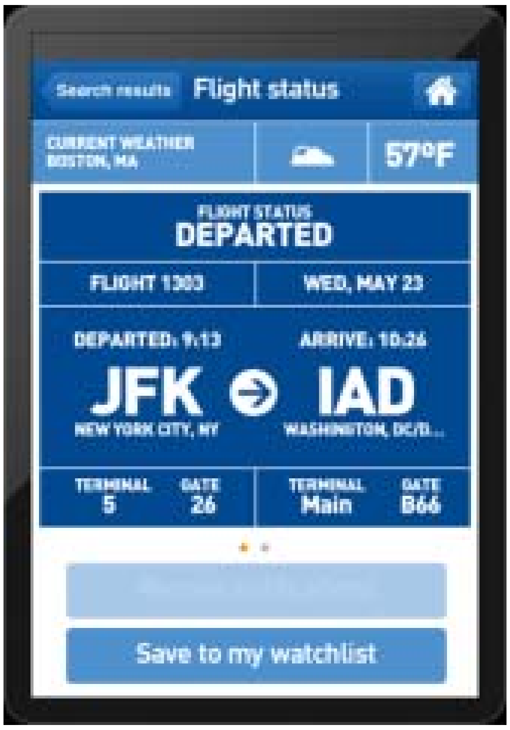


Figure 2. Flight status updates [2].

Flight and travel status module key features:

- Fully customizable workflows, business rules and events
- Granular events control
- Tailors alerts for travelers, subscribers and staff
- Uses real-time event triggers
- Uses airline FLIFO feed for accurate real-time movement data
- Delivery via APP Push, Email, SMS and Voice
- Supports vouchers, advertisements and travel advice

The benefit for an airline:

- Harmonize airline communications across all touch points
- Reduce traveler and staff stress
- Provide targeted and timely information
- Optimize airline customers experience
- Improve ground efficiency and on-time performance

Queue manager

On the reservation system PNRs are placed on queues. Usually different queue means different processing workflow, like sending different notification for the PNR, or PNRs those have accepted schedule change and those declined schedule change.

Notification queues have never been so efficiently organized. Queue Manager works behind the scenes to give airline a smooth new way of automatically sorting, filtering and categorizing your PNRs for simple and fast processing. It lets airline to highlight specific queues, create and automate complicated tasks that can take hours for airline staff to deal with manually.

By interfacing directly with any reservation system, it can show airline at a glance which passengers have responded or acted on a notification and what they've done. It then feeds this back into airline reservation system, triggering changes in PNR data which can in turn trigger a change in queue or a new process, such as reissuing a ticket or responding to a Call Back request.

Queue Manager is compatible with all our other 15below passenger notification modules [2].

Queue manager module key features:

- Sort, filter and categorize PNRs
- Perform complex tasks such as re-accommodation, revalidation and reissues
- Hold and schedule notifications sends
- Automatically clean queues
- Trigger actions with PNR data and changes
- Setup and manage tasks with complex workflows
- Full reporting capabilities

The benefit for an airline:

- Reduce manual handling, time and costs
- Enhance queue visibility
- Optimize operational efficiency
- Support all other 15below passenger communications

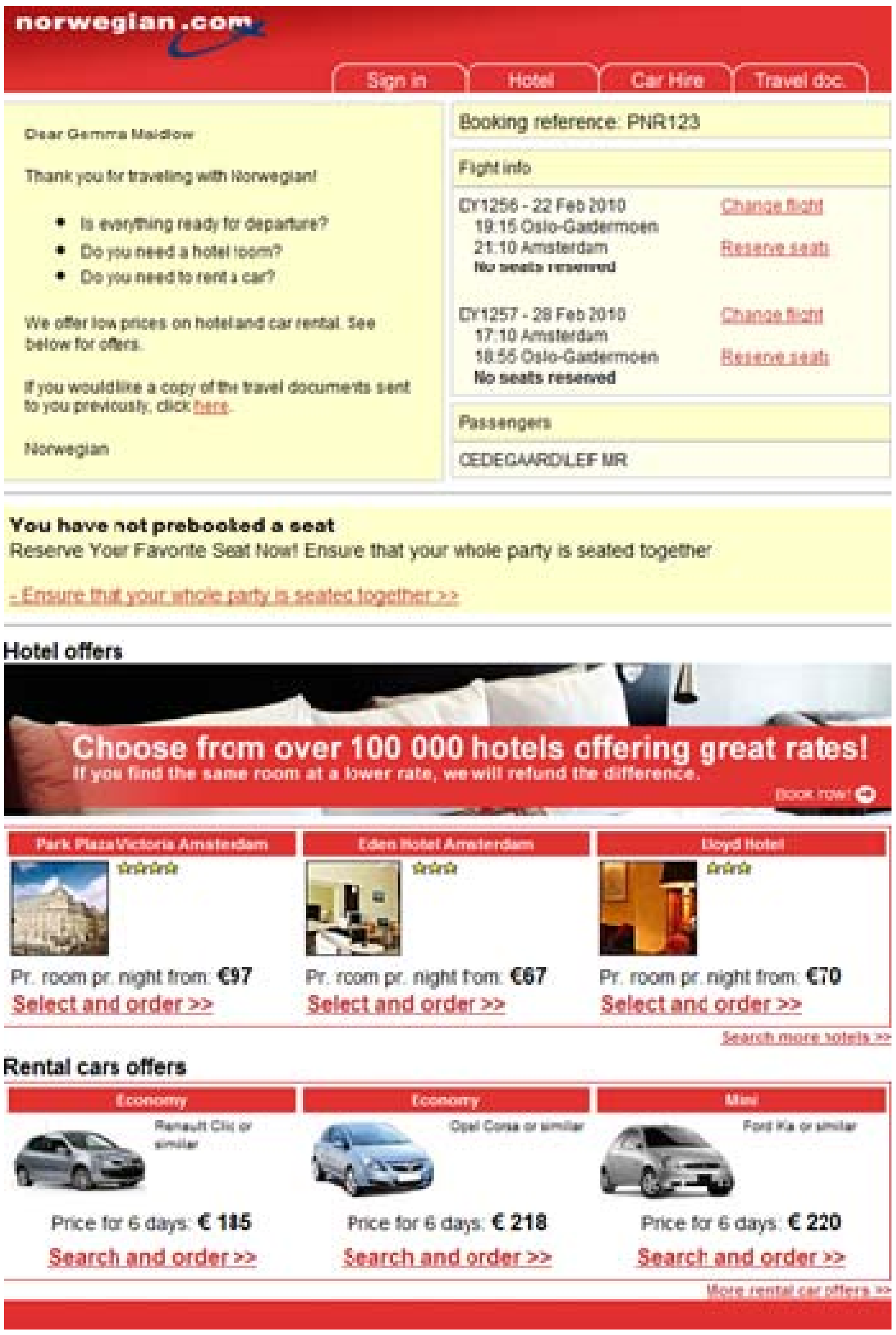
Pre-departure communications

The main idea of this module is to remind, confirm and promote. Itinerary-based notifications with targeted advertising [Figure 3].

Delight airline passengers with all the practical journey information and the added extras they need to plan the perfect trip. There's no simpler way of giving them peace-of-mind before their journey and driving up airline revenue at the same time.

This module interfaces directly with airline reservation system, schedules and third party feeds. Airline can use it to improve their passengers' travel experience before they even walk out their front doors.

Details such the weather, public transport information or top attractions make it easier for them to decide on what to pack and book before they go, while special offers or real-time quotes from airline partners can be targeted according to details such as their class of travel, departure date or destination.



norwegian.com

Sign in | Hotel | Car Hire | Travel doc.

Dear Gemma Maidlow

Thank you for traveling with Norwegian!

- Is everything ready for departure?
- Do you need a hotel room?
- Do you need to rent a car?

We offer low prices on hotel and car rental. See below for offers.

If you would like a copy of the travel documents sent to you previously, click here.

Norwegian

Booking reference: PNR123

Flight info

DY1256 - 22 Feb 2010
19:15 Oslo-Gardermoen
21:10 Amsterdam
No seats reserved
Change flight
Reserve seats

DY1257 - 28 Feb 2010
17:10 Amsterdam
18:55 Oslo-Gardermoen
No seats reserved
Change flight
Reserve seats

Passengers

OEDEGAARD LEIF MR

You have not prebooked a seat
Reserve Your Favorite Seat Now! Ensure that your whole party is seated together

- Ensure that your whole party is seated together >>

Hotel offers

Choose from over 100 000 hotels offering great rates!
If you find the same room at a lower rate, we will refund the difference.
Book now!

Park Plaza Victoria Amsterdam	Eden Hotel Amsterdam	Lloyd Hotel
Pr. room pr. night from: €97	Pr. room pr. night from: €67	Pr. room pr. night from: €70
Select and order >>	Select and order >>	Select and order >>

Search more hotels >>

Rental cars offers

Economy	Economy	Mini
Renault Clio or similar	Opel Corsa or similar	Ford Ka or similar
Price for 6 days: € 185	Price for 6 days: € 218	Price for 6 days: € 220
Search and order >>	Search and order >>	Search and order >>

More rental car offers >>

Figure 3. Travel reminder [2].

Pre-departure communications module key features:

- Flight and other travel confirmations
- Integrated travel advisories, weather reports and destination-specific information
- Travel checklist – such as visa, insurance, baggage allowance
- Targeted ancillary promotions (including availability)
- Other content enrichment – including maps, social intelligence, traffic, entertainment.
- Extreme personalization

The benefit for an airline:

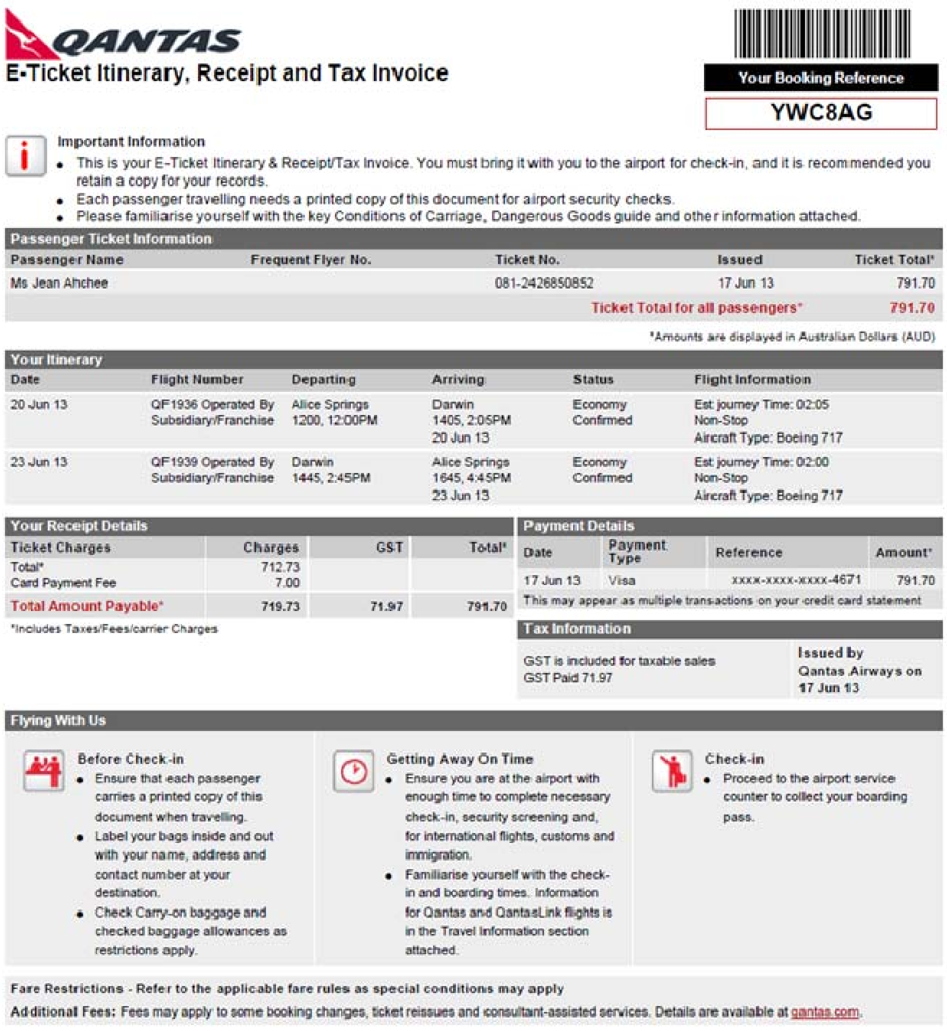
- Reduce inbound calls
- Personalize and enhance airline customers experience
- Up sell and increase revenue streams
- Strengthen airline understanding and knowledge of their customers

Itineraries, booking confirmations and receipts

For many travelers, the itinerary, confirmation and receipts they get from airline are their first taste of the airline brand [Figure 4]. They're the perfect opportunity to make a memorable first impression and offer to passengers personalized service and targeted ancillary products.

This module interfaces with airline reservation and finance systems to automatically send practical booking information directly to your passengers via Email, SMS, App Push or print. It is also possible to add iCal, PDF and Passbook attachments to deliver a seamless customer experience.

No matter which format passenger chooses, it is possible to pull in content from other systems that is relevant to their destination or travel dates, including offers for airline ancillary products or those of airline partners. Each confirmation and receipt presents an airline with another opportunity to sell and impress.



E-Ticket Itinerary, Receipt and Tax Invoice

Your Booking Reference

YWC8AG

Important Information

- This is your E-Ticket Itinerary & Receipt/Tax Invoice. You must bring it with you to the airport for check-in, and it is recommended you retain a copy for your records.
- Each passenger travelling needs a printed copy of this document for airport security checks.
- Please familiarise yourself with the key Conditions of Carriage, Dangerous Goods guide and other information attached.

Passenger Ticket Information

Passenger Name	Frequent Flyer No.	Ticket No.	Issued	Ticket Total*
Ms Jean Ahchee		081-2426850852	17 Jun 13	791.70
			Ticket Total for all passengers*	791.70

*Amounts are displayed in Australian Dollars (AUD)

Your Itinerary

Date	Flight Number	Departing	Arriving	Status	Flight Information
20 Jun 13	QF1936 Operated By Subsidiary/Franchise	Alice Springs 1200, 12:00PM	Darwin 1405, 2:05PM 20 Jun 13	Economy Confirmed	Est journey Time: 02:05 Non-Stop Aircraft Type: Boeing 717
23 Jun 13	QF1939 Operated By Subsidiary/Franchise	Darwin 1445, 2:45PM	Alice Springs 1645, 4:45PM 23 Jun 13	Economy Confirmed	Est journey Time: 02:00 Non-Stop Aircraft Type: Boeing 717

Your Receipt Details

Ticket Charges	Charges	GST	Total*
Total*	712.73		
Card Payment Fee	7.00		
Total Amount Payable*	719.73	71.97	791.70

*Includes Taxes/Fees/carrier Charges

Payment Details

Date	Payment Type	Reference	Amount*
17 Jun 13	Visa	xxxx-xxxx-xxxx-4671	791.70

This may appear as multiple transactions on your credit card statement

Tax Information

GST is included for taxable sales
GST Paid 71.97

Issued by Qantas Airways on 17 Jun 13

Flying With Us

Before Check-in
- Ensure that each passenger carries a printed copy of this document when travelling.
- Label your bags inside and out with your name, address and contact number at your destination.
- Check Carry-on baggage and checked baggage allowances as restrictions apply.

Getting Away On Time
- Ensure you are at the airport with enough time to complete necessary check-in, security screening and, for international flights, customs and immigration.
- Familiarise yourself with the check-in and boarding times. Information for Qantas and QantasLink flights is in the Travel Information section attached.

Check-in
- Proceed to the airport service counter to collect your boarding pass.

Fare Restrictions - Refer to the applicable fare rules as special conditions may apply

Additional Fees: Fees may apply to some booking changes, ticket reissues and consultant-assisted services. Details are available at qantas.com.

Figure 4. Travel itinerary [2].

Itineraries, booking confirmations and receipts module key features:

- Sophisticated business rules and fare calculations
- Barcode support
- Attachment support (including iCal and PDF)
- Content enrichment – including baggage allowance, flight duration, maps, weather
- Real-time ancillary partner integration
- SMS confirmations

The benefit for an airline:

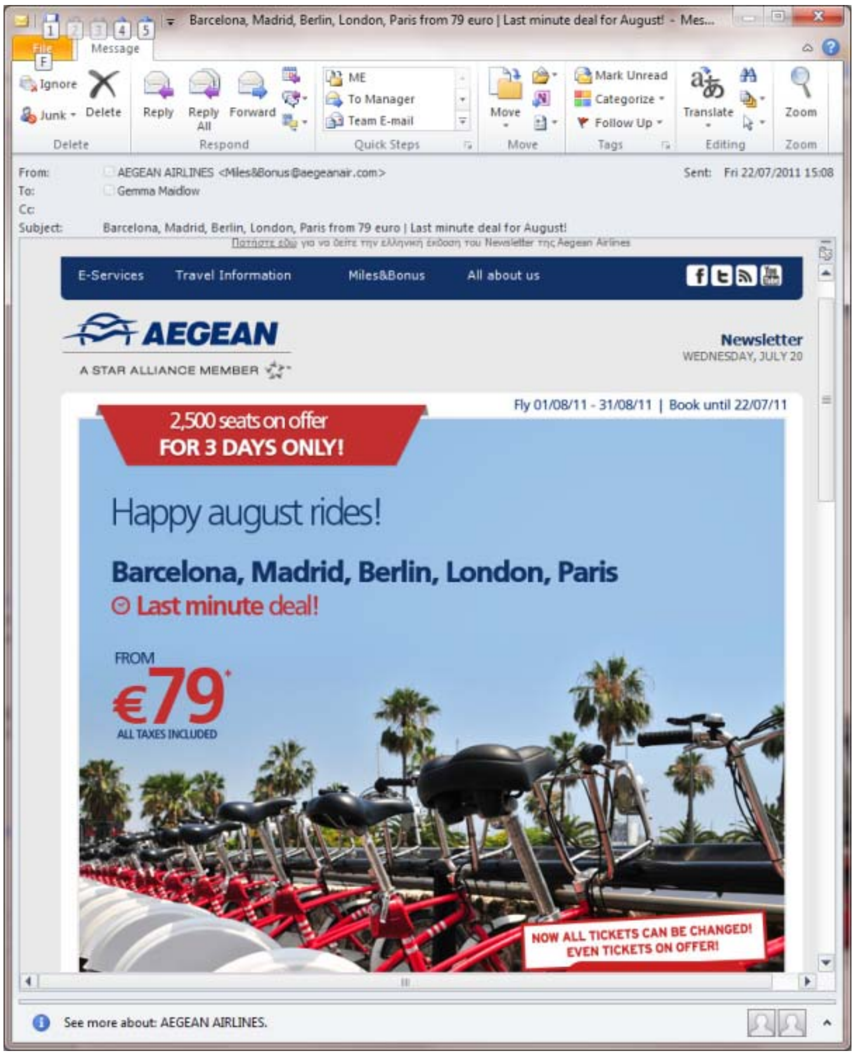
- Improve traveler experience
- Generate additional revenue
- Bring brand consistency across multiple systems and offices
- Automate tax and legal obligations

Frequent flyer and loyalty

Good long-distance relationships are built through communication. By interfacing with airline frequent flyer database, our Frequent Flyer and Loyalty module makes it simple for airline to send their customers personalized offers and information in the format they prefer [Figure 5].

From the moment a person joins airline programme an airline can communicate in a way that is direct, personal and relevant to them. Airline can stay connected to them between journeys, and build reputation for great service even when they're not travelling.

Being a two-way system, the module also cuts down on the manual work airline team needs to do to maintain their loyalty programme database. It automatically updates airline customers' subscription choices and cleans the database as it communicates [2].



Barcelona, Madrid, Berlin, London, Paris from 79 euro | Last minute deal for August! - Mes...

From: AEGEAN AIRLINES <Miles&Bonus@aegeanair.com>
Sent: Fri 22/07/2011 15:08
To: Gemma Maidlow
Cc:
Subject: Barcelona, Madrid, Berlin, London, Paris from 79 euro | Last minute deal for August!

Πατήστε εδώ για να δείτε την ελληνική έκδοση του Newsletter της Aegean Airlines

Figure 5. Special offer email [2].

Frequent flyer and loyalty module key features:

- Hosted or integrated with Frequent Traveler database
- Preference-based notifications for travelers
- Management of Frequent Flyer or loyalty program
- Tier-based offers and invitation notifications
- Dynamic real-time subscription management
- List hygiene management

The benefit for an airline:

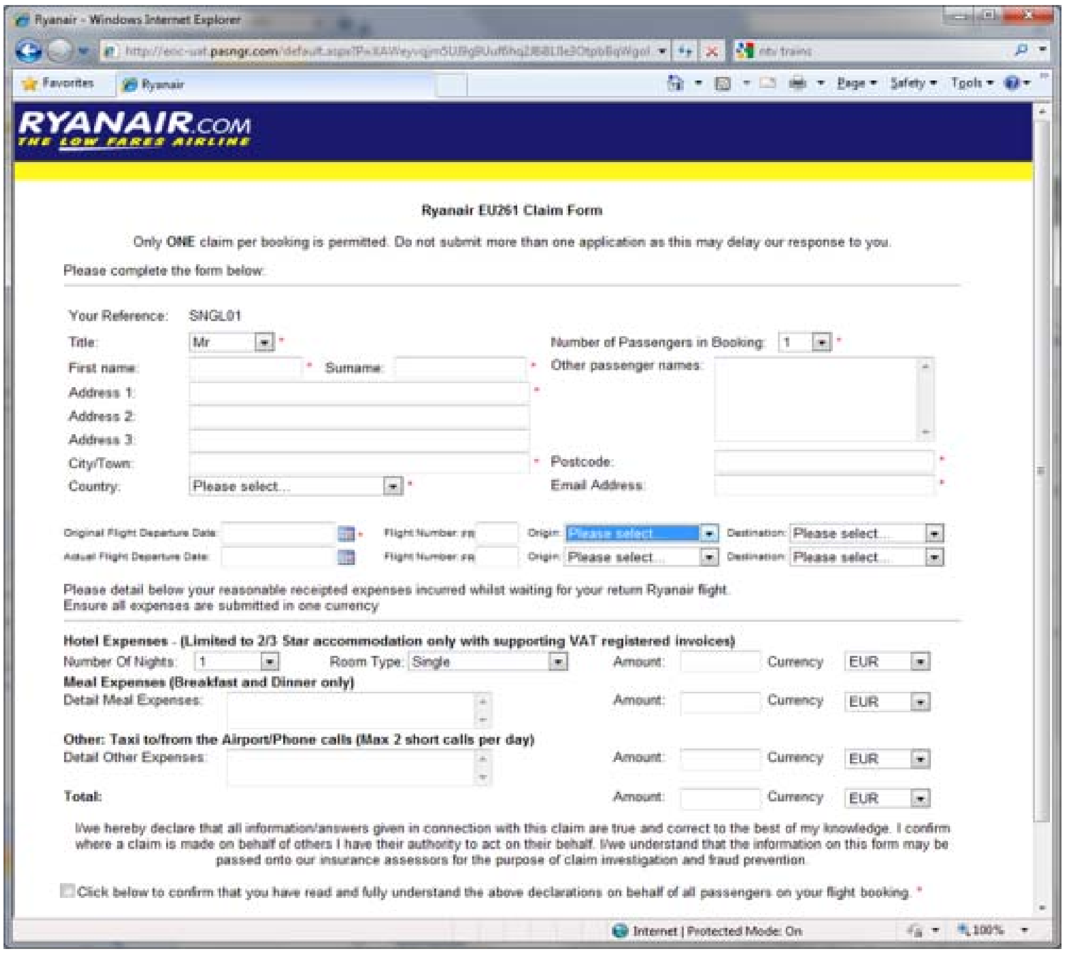
- Drive loyalty program sign-up and on-going customer engagement
- Build brand loyalty with highly personalized service
- Drive sales and revenue with targeted promotions
- Clean and maintain database lists

Special service handling

Reduce manual handling and processing of travelers needing some extra attention, by automating all those ‘little’ things that take up time and resource.

By interfacing with airline reservation system, this tool can help airline teams do much more with less effort - from gathering advance passenger information to communicating with groups or making sure lost baggage is reunited with its owner.

Send updates or requests to a filtered list of passengers in a couple of clicks. Passengers’ replies or actions are automatically fed back into airline reservation system, triggering the next step in the process, meaning staff can quickly identify who needs to be contacted directly and focus their effort where it is needed [2].


Ryanair EU261 Claim Form

Only ONE claim per booking is permitted. Do not submit more than one application as this may delay our response to you.

Please complete the form below:

Your Reference:	SNGL01		
Title:	Mr *	Number of Passengers in Booking:	1 *
First name:	* Surname: *	Other passenger names:	
Address 1:	*		
Address 2:			
Address 3:			
City/Town:	*	Postcode:	*
Country:	Please select... *	Email Address:	*

Original Flight Departure Date: *	Flight Number: FR	Origin: Please select...	Destination: Please select...
Actual Flight Departure Date:	Flight Number: FR	Origin: Please select...	Destination: Please select...

Please detail below your reasonable receipted expenses incurred whilst waiting for your return Ryanair flight.
Ensure all expenses are submitted in one currency

Hotel Expenses - (Limited to 2/3 Star accommodation only with supporting VAT registered invoices)

Number Of Nights: 1 Room Type: Single Amount: Currency EUR

Meal Expenses (Breakfast and Dinner only)

Detail Meal Expenses: Amount: Currency EUR

Other: Taxi to/from the Airport/Phone calls (Max 2 short calls per day)

Detail Other Expenses: Amount: Currency EUR

Total: Amount: Currency EUR

I/we hereby declare that all information/answers given in connection with this claim are true and correct to the best of my knowledge. I confirm where a claim is made on behalf of others I have their authority to act on their behalf. I/we understand that the information on this form may be passed onto our insurance assessors for the purpose of claim investigation and fraud prevention.

☐ Click below to confirm that you have read and fully understand the above declarations on behalf of all passengers on your flight booking. *

Figure 6. Ryanair EU261 Claim form screenshot [2].

Special service handling module key features:

- Group travel processing
- APIS, ESTA and secure flights
- Unaccompanied minor documentation
- EU261 expense applications [Figure 6]
- SSR related workflows
- Lost, delayed and damaged baggage

The benefit for an airline:

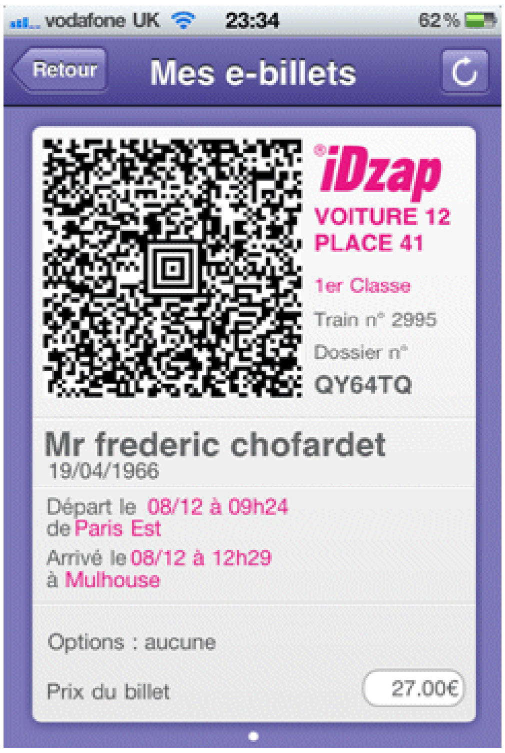
- Reduce manual processing and call centre volume
- Manage airport queues
- Optimize operational efficiency
- Improve customer experience

Ticketing

Convenience and speed are two things about instant tickets that customers love. By letting customers get their tickets on the move or in their homes airline reduce potential stress at the start of their journey and make life easier for airline staff too.

15below Ticketing module integrates with airline bookings system and generates tickets in the format and language to suit airline customer, whether they're on their phone, tablet or PC [Figure 7].

You can add targeted content to the tickets too, including anything from real-time hotel quotes for their destination, to special offers at cafés or kiosks along their route [2].


vodafone UK 23:34 62 %

Retour Mes e-billets

iDzap
VOITURE 12
PLACE 41
1er Classe
Train n° 2995
Dossier n°
QY64TQ

Mr frederic chofardet
19/04/1966

Départ le 08/12 à 09h24
de Paris Est
Arrivé le 08/12 à 12h29
à Mulhouse

Options : aucune
Prix du billet 27.00€

Figure 7. E-Ticket on Smartphone [2].

Ticketing module key features:

- Real-time ticket production
- 1D and 2D barcodes
- Ancillary promotions
- Third party content enrichment such as maps
- Multiple channels – email, PDF, SMS and App delivery

The benefit for an airline:

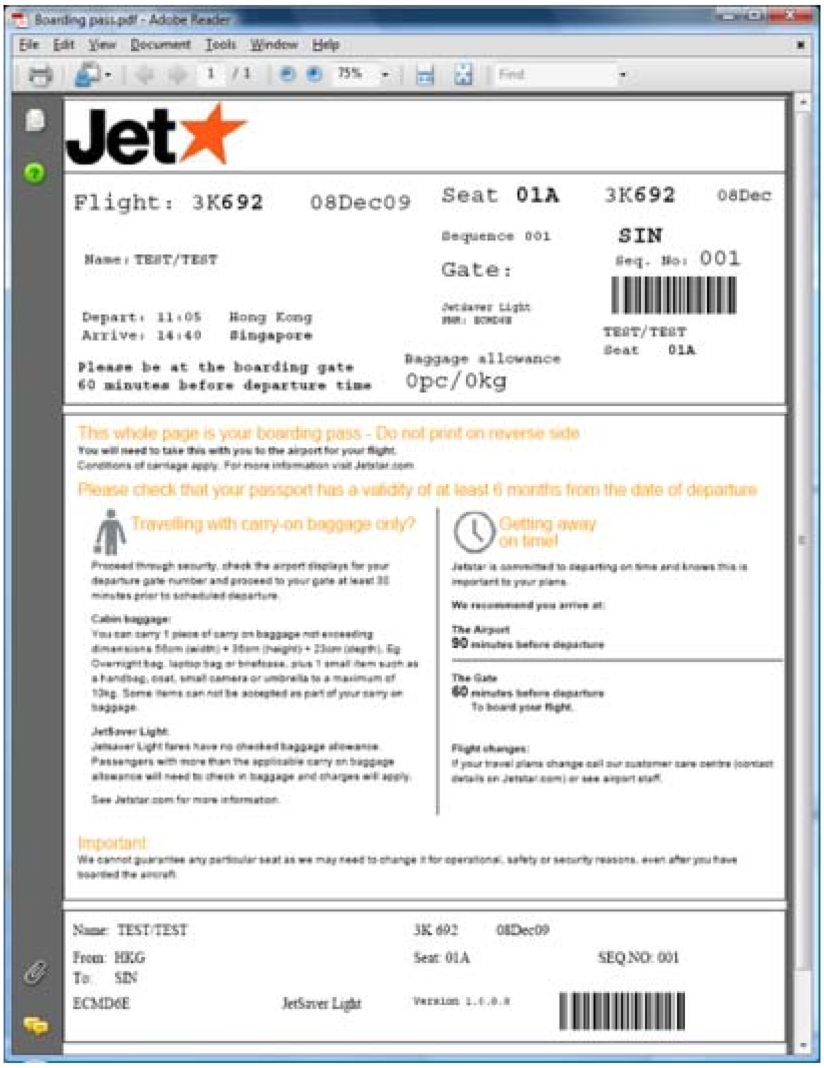
- Simplify back-office processes
- Minimize manual intervention
- Reduce queues at station
- Enhance traveler experience

Check in and boarding passes

There's no need for airline or passengers to live in dread of long airport queues. With online check-in, and boarding passes that are sent straight to passenger's mobile, computer or printer the start of a journey can be a smoother experience for everyone.

15below automated check-in module interfaces with airline DCS and reservation system and can also hook into partner feeds, letting airline send their passengers both their boarding pass and personalized offers or useful information for their airport or destination [Figure 8].

The system can also filter out passengers who aren't eligible for online check in, automatically sending them the information they need for the start of their journey [2].


Boarding pass.pdf - Adobe Reader

Jet★

Flight: 3K692 08Dec09
Seat 01A
3K692 08Dec

Sequence 001
SIN

Name: TEST/TEST
Gate:
Seq. No: 001

JetSaver Light
PNR: ECMD6E

Depart: 11:05 Hong Kong
Arrive: 14:40 Singapore

TEST/TEST
Seat 01A

Please be at the boarding gate
60 minutes before departure time

Baggage allowance
0pc/0kg

This whole page is your boarding pass - Do not print on reverse side
You will need to take this with you to the airport for your flight.
Conditions of carriage apply. For more information visit Jetstar.com

Please check that your passport has a validity of at least 6 months from the date of departure

Travelling with carry-on baggage only?

Proceed through security, check the airport displays for your departure gate number and proceed to your gate at least 30 minutes prior to scheduled departure.

Cabin baggage:
You can carry 1 piece of carry on baggage not exceeding dimensions 56cm (width) + 36cm (height) + 23cm (depth). Eg Overnight bag, laptop bag or briefcase, plus 1 small item such as a handbag, coat, small camera or umbrella to a maximum of 10kg. Some items can not be accepted as part of your carry on baggage.

JetSaver Light:
JetSaver Light fares have no checked baggage allowance. Passengers with more than the applicable carry on baggage allowance will need to check in baggage and charges will apply.

See Jetstar.com for more information.

Getting away on time!

Jetstar is committed to departing on time and knows this is important to your plans.

We recommend you arrive at:

The Airport
90 minutes before departure

The Gate
60 minutes before departure
To board your flight.

Flight changes:
If your travel plans change call our customer care centre (contact details on Jetstar.com) or see airport staff.

Important
We cannot guarantee any particular seat as we may need to change it for operational, safety or security reasons, even after you have boarded the aircraft.

Name: TEST/TEST
3K 692 08Dec09
From: HKG
Seat: 01A
SEQ NO: 001
To: SIN
ECMD6E
JetSaver Light
Version 1.0.0.0

Figure 8. Boarding pass [2].

Check in and boarding passes module key features:

- Fully automated
- IATA standard barcode support
- Multi-channel support including PDF and Mobile boarding pass
- Filter ineligible passengers (e.g. unaccompanied minor)
- Content enrichment (e.g. maps, traffic and more)
- Full business intelligence and reporting capabilities

The benefit for an airline:

- Reduce queues and minimize disruption
- Enhance passenger experience and reduce stress
- Improve ground operations and reduce costs
- Increase ancillary revenue

All of these modules are used with current 15below platform communication channels – Email, SMS, Voice, Fax, Print and Push.

2.3 COMMUNICATION CHANNELS

15below current platform PASNGR supports most popular send channels across the world. However, some of them are more popular in one region and some of them are more popular in other. That is why it is important to support as many channels as possible to stay competitive in all regions.

Email

Email is the most common send channel. It is very flexible as using HTML it is possible to add images and styling to make a branded and nice looking email. Email also supports attachments, so it makes possible to add PDF, iCalendar, Passbook or any other attachment.

However to receive an email it is necessary to have internet connection, so it might be expensive to use while traveling abroad, because of nicely formatted Email file size. As an extra come all images and attachments. Another downside of email is formatting support. Not all devices and readers support email formatting or supports only partially, making the email look broken.

SMS

This send channel is popular in East Asia and Pacific. It is simple, reliable and costs nothing to receive, what makes it to be a very good option while traveling abroad.

However it is an expensive option for airline. Furthermore a single SMS is limited to 160 characters, so the amount of text is very limited and to save the characters count abbreviations are used very frequently, what can lead to the SMS which is hard to understand.

Voice

Automated call, when a “robot” is calling you is the Voice channel. It is popular in the United States. It is also free to receive. Also interaction is available by pressing keys on request.

The downside of this channel is cost of calls. Also, because the message is auto generated, it usually has no intonation. Finally the call is not saved anywhere after it is finished. This might be a problem when the message was not properly understood or there was a mobile network coverage problem and the voice was cracking.

Fax

Companies, or sometimes individuals, might prefer to get their notifications printed by their fax machine. Fax also supports formatting, so it is possible to make notification look nicer.

However usage of fax is being replaced by emails, so the number of users keeps reducing. Another problem is formatting. Fax usually prints notification in black and white, so colorful text or banners are not an option for this send channel.

Print

Some of airlines prefer to get every notification printed by their printer as an extra, for various purposes. PASNGR can have mapped any network printer, event from external network, and print a copy any notification.

This send channel is used only by airlines themselves as it would be next to impossible to configure each passenger's printer to receive and print a notification.

Push

Push notifications is an alternative of to SMS. It alerts the customer as soon as the notification is received on device. It can also contain payload data for preferred Smartphone application.

The problem with this send channel it is using internet connection and is supported only by some of Smart phones and tablets. Furthermore the amount of payload data is limited, so it is recommended to serialize the data before sending and de-serialize after receiving, but de-serializing mean having an app which would support that.

2.4 CASE STUDY

It is already clear, that current 15below communications platform has 9 modules. As a case study, there will be analyzed only one of them. I have chosen to analyze the Disruption module. The reason of the choice is its popularity – almost every 15below customer is using it. This is not a surprise knowing that it helps to control major disruptions and save money on call center in the same time.

Major disruptions usually are caused by extreme weather conditions. For example Hurricane Sandy resulted in a total of 20,254 flight cancellations in North America between October 27 and November 1, 2012, as well as a significant loss of revenue and profits for the major U.S. carriers [3]. The hurricane cost carriers huge amounts of money [Table 1].

Airline	Flights cancelled	Lost revenue	Lost profit	Revenue/flight	Profit/flight
American	759	$65M	N/A	$85,000	N/A
US Airways	1,454	N/A	$20M	N/A	$14,000
Delta	1,293	$75M	$45M	$58,000	$35,000
United	2,149	$140M	$35M	$65,000	$16,000

Table 1. The impact of Hurricane Sandy [3].

These figures could be different if passengers' re-accommodation was handled not by call centers, but by automated platform. So how does the disruption module work?

Lets say some irregular operation (IROP) has happened on the airline side. From that point all the process goes to 15below side [Figure 9].

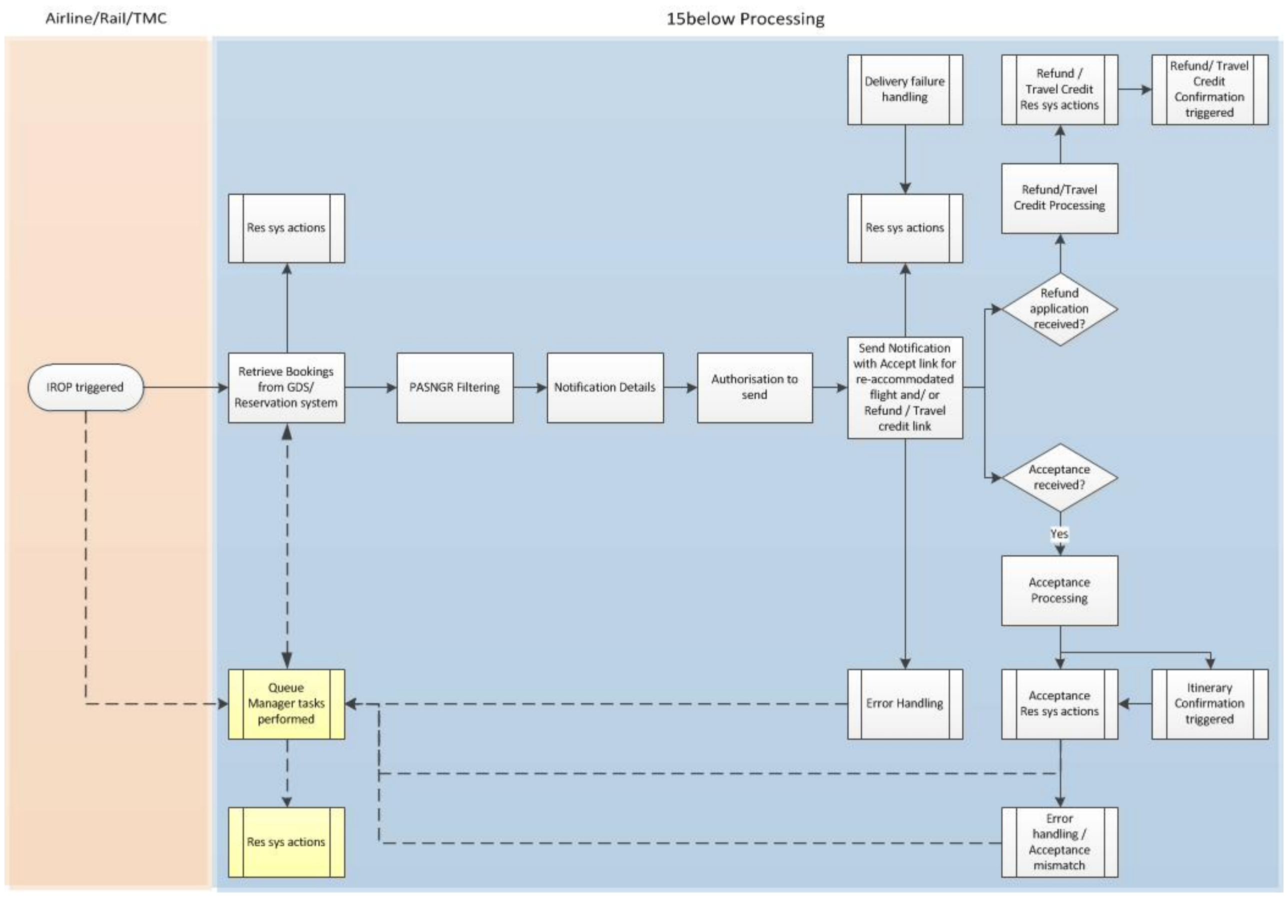
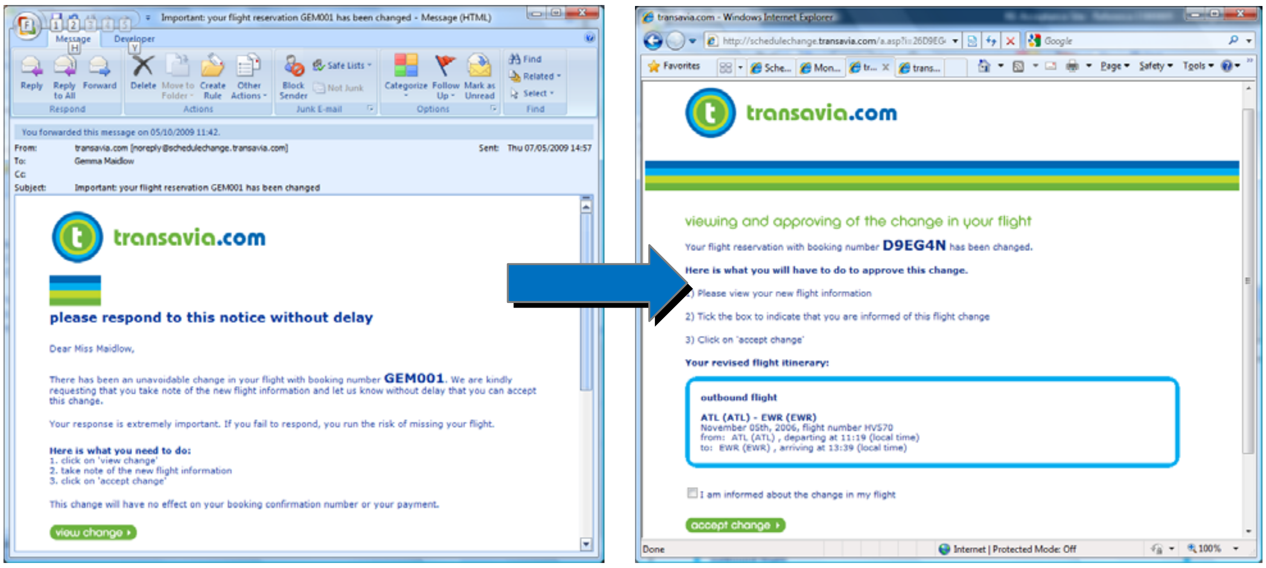

Airline/Rail/TMC
15below Processing
IROP triggered
Retrieve Bookings from GDS/ Reservation system
Res sys actions
PASNGR Filtering
Notification Details
Authorisation to send
Send Notification with Accept link for re-accommodated flight and/ or Refund / Travel credit link
Delivery failure handling
Res sys actions
Refund application received?
Refund/Travel Credit Processing
Refund / Travel Credit Res sys actions
Refund/ Travel Credit Confirmation triggered
Acceptance received?
Yes
Acceptance Processing
Acceptance Res sys actions
Itinerary Confirmation triggered
Error Handling
Error handling / Acceptance mismatch
Queue Manager tasks performed
Res sys actions

Figure 9. 15below Disruption module workflow [2].

The very first step for 15below would be to retrieve affected bookings from GDS or other Reservation system. After retrieving bookings it needs updating them with the status back in GDS or Reservation system. Furthermore if bookings have to be placed on a different queue, Queue Manager Module must be used as well. The next step is bookings filtering. 15below communication platform has filters such as flight date, flight number, language etc. Then it needs to choose the right notification template and send channel. After authorization notifications are sent to filtered recipients. In addition to this status update can be applied to the original booking in GDS or Reservation System with either success or failure. In case of failure, using Queue Manager, the booking can be placed on a different queue.

The notification had a link to either Refund or Acceptance microsite, where customer could take necessary actions [Figure 10]. Those actions are fed back to GDS or Reservation System, so the airline could see it at take necessary further actions.



You forwarded this message on 05/10/2009 11:42.

From: transavia.com [noreply@schedulechange.transavia.com] Sent: Thu 07/05/2009 14:57
To: Gemma Maidlow
Cc:
Subject: Important: your flight reservation GEM001 has been changed

please respond to this notice without delay

Dear Miss Maidlow,

There has been an unavoidable change in your flight with booking number **GEM001**. We are kindly requesting that you take note of the new flight information and let us know without delay that you can accept this change.

Your response is extremely important. If you fail to respond, you run the risk of missing your flight.

Here is what you need to do:
1. click on 'view change'
2. take note of the new flight information
3. click on 'accept change'

This change will have no effect on your booking confirmation number or your payment.

view change

viewing and approving of the change in your flight

Your flight reservation with booking number **D9EG4N** has been changed.

Here is what you will have to do to approve this change.

1) Please view your new flight information

2) Tick the box to indicate that you are informed of this flight change

3) Click on 'accept change'

Your revised flight itinerary:

outbound flight

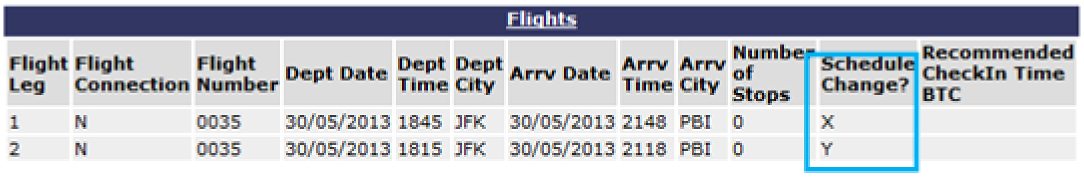
ATL (ATL) - EWR (EWR)
November 05th, 2006, flight number HV570
from: ATL (ATL) , departing at 11:19 (local time)
to: EWR (EWR) , arriving at 13:39 (local time)

☐ I am informed about the change in my flight

accept change

Figure 10. Schedule change acceptance [2].

The status could also be seen in the 15below messaging platform user interface [Figure 11].



Flights											
Flight Leg	Flight Connection	Flight Number	Dept Date	Dept Time	Dept City	Arrv Date	Arrv Time	Arrv City	Number of Stops	Schedule Change?	Recommended CheckIn Time BTC
1	N	0035	30/05/2013	1845	JFK	30/05/2013	2148	PBI	0	X	
2	N	0035	30/05/2013	1815	JFK	30/05/2013	2118	PBI	0	Y	

Figure 11. Booking status stored in PASNGR [2].

2.5 RESULTS

After current 15below platform PASNGR analysis is done, I can sum up which of modules could be replicated in 15below mobile application.

As a core function I would suggest Ticketing, so the user could store his tickets on his mobile device and have them wherever he goes. In addition to this Disruption module could be integrated with ticket and provide any disruption information to the traveler promptly. Schedule change acceptance should go as a part of functionality, so the user could accept new flight with one button click. Also Survey module could be integrated as a part of post trip customer satisfaction feedback.

3. Reservation system analysis

3.1 INTRODUCTION

After current 15below platform analysis is done, I have fund that a crucial point for the system to work is data retrieve. The data always comes from third party feeds. In most cases the source is Global Distribution Systems as known as Reservation Systems. The goal of this chapter is to take a deeper look into one of GDS, to see how it works, what data it gives. This is very important step as it gives a view of what the platform can retrieve and what the product can come up from it as a result.

There are a number of GDS available to choose from. However they have the same purpose and they all work in very similar way, so I decided to do analysis on only one GDS, which is a major one in the global market and is called Amadeus.

3.2 GLOBAL DISTRIBUTION SYSTEMS

Amadeus is a transaction processor for the global travel and tourism industry, providing transaction processing power and technology solutions to both travel providers (including full service carriers and low-cost airlines, hotels, rail operators, cruise and ferry operators, car rental companies and tour operators) and travel agencies (both online and offline). The company acts both as a worldwide network connecting travel providers and travel agencies through a processing platform for the distribution of travel products and services, and as a provider of a portfolio of IT solutions which automate certain mission - critical business processes, such as reservations, inventory management and operations for travel providers [Figure 12]. [4]

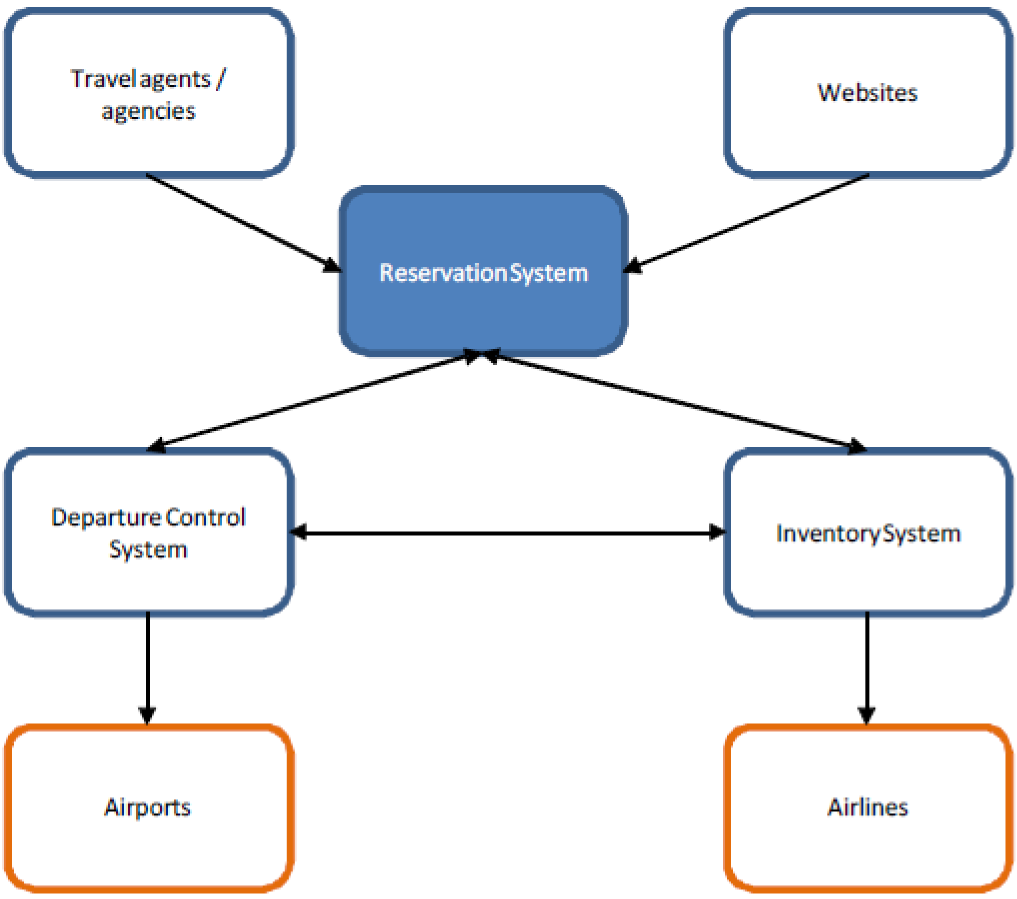


Figure 12. Schema of the main business areas of Amadeus SAS. [4]

This figure gives a clear view of how GDS really works. Although this schema represents Amadeus business areas, it is very familiar to any other GDS. This figure also shows other fact about GDS – it centralized system. It might be fed and red by different parties.

Travel agencies and websites, obviously airline websites, could be grouped together as they do very similar thing – place bookings into Reservation system. Now Departure Control System and Inventory System is the area where 15below acts with their products. For example Flight Status depends on Departure Control System feed and in the same time the product is a part of an Airline inventory.

In 15below most of the data is about passengers and their itineraries, so I would rather focus more on this area.

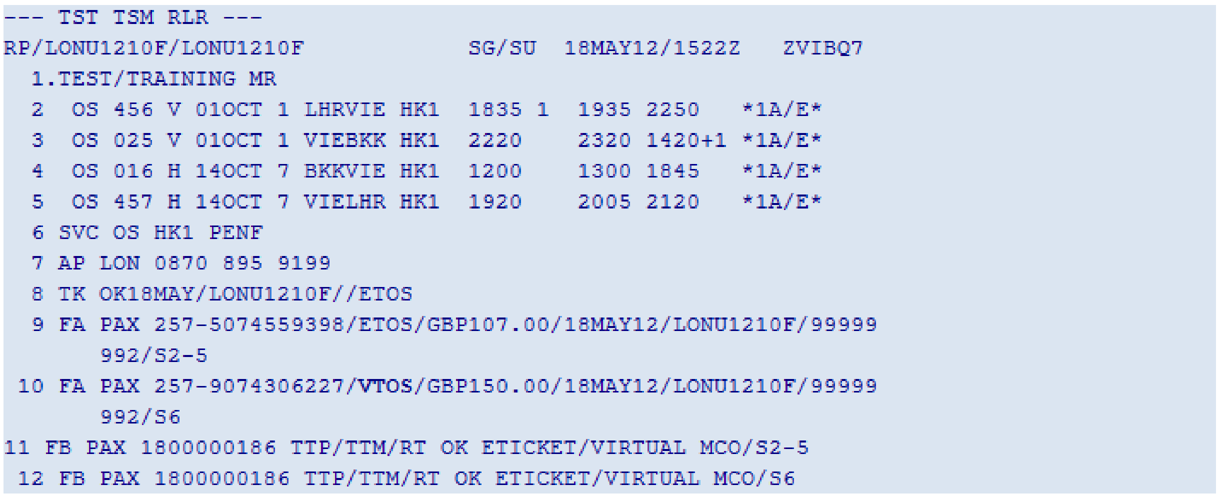
If you ever travelled on plane, you definitely must have seen a Booking Reference. It is a six characters long code, which contains only Latin letters or numbers. In travel industry this record is called Passenger Name Record (PNR). PNR is a record in the database of a Computer Reservation System (CRS) that contains the itinerary for a passenger, or a group of passengers travelling together. [4]

PNR is created when a passenger books an itinerary with airline website or travel agent. The newly created PNR is placed in CRS used by an airline or travel agent. This is typically one of the large Global Distribution Systems, such as Amadeus, Sabre, Worldspan or Galileo, but if the booking is made directly with an airline the PNR can also be in the database of the airline's CRS.

When portions of the travel are not provided by the holder of the Master PNR, then copies of the PNR information are sent to the CRSes of the airlines that will be providing transportation. These CRSes will open copies of the original PNR in their own database to manage the portion of the itinerary for which they are responsible. [4]

The main problem with PNR – it is not unique. IT is even not unique within same airline. Of course it is very low chance to get the same PNR reference at the same time, having in mind how many combinations it is possible to get, but the fact still exists. To be more accurate, sometimes a combination of PNR and other detail, such as origin, destination, flight date, passenger last name or something else is used. This helps to reduce the chance of wrong itinerary to be retrieved.

Another problem is the way information is stored – displayed. In general it is a free text information [Figure 13]. On the other hand it is structured as a text and knowing which bit represents which data it is possible to parse the information into computer readable format, like XML or something else. The reason behind that is GDS systems date back to 1960, when computers were rarely used for data manipulations and all stored info was structured enough for reading by human. Also every GDS might have a different format and different identifiers, what will result a different parser per GDS.



```
--- TST TSM RLR ---
RP/LONU1210F/LONU1210F            SG/SU  18MAY12/1522Z   ZVIBQ7
  1.TEST/TRAINING MR
  2  OS 456 V 01OCT 1 LHRVIE HK1  1835 1  1935 2250   *1A/E*
  3  OS 025 V 01OCT 1 VIEBKK HK1  2220    2320 1420+1 *1A/E*
  4  OS 016 H 14OCT 7 BKKVIE HK1  1200    1300 1845   *1A/E*
  5  OS 457 H 14OCT 7 VIELHR HK1  1920    2005 2120   *1A/E*
  6 SVC OS HK1 PENF
  7 AP LON 0870 895 9199
  8 TK OK18MAY/LONU1210F//ETOS
  9 FA PAX 257-5074559398/ETOS/GBP107.00/18MAY12/LONU1210F/99999
       992/S2-5
 10 FA PAX 257-9074306227/VTOS/GBP150.00/18MAY12/LONU1210F/99999
       992/S6
11 FB PAX 1800000186 TTP/TTM/RT OK ETICKET/VIRTUAL MCO/S2-5
 12 FB PAX 1800000186 TTP/TTM/RT OK ETICKET/VIRTUAL MCO/S6
```

Figure 13. Amadeus PNR example. [6]

Most airlines typically parse data from the PNR into separate files to populate operational and analytical data marts based on departmental needs. For example, revenue management receives the number of bookings per flight, revenue accounting receives ticketing data and frequent flyer receives flown data at the segment level, and so on. [5] This is also a point of interest of 15below.

3.3 CASE STUDY

Customer needs are endless. They always want more, but often they are not getting what they want. The main thing which decides if customers will take what they want is how easy they can get it. This is a key factor for economy to spin around. If you want to stay in the business, you must be as dynamic as customer's needs are, or even more. That's why offering a narrow choice of services is dangerous. Customers often want not just a flight, but they want to get the whole package – flight, hotel, car, etc – in one go. If an airline won't sell the package, there will be something else that will do and even possibly will do this cheaper. This became a real challenge in recent years. This has forced airlines to cut costs and look for new revenue streams in order to stay in business. In the midst of this transformation, ancillary revenues have come into the spotlight, giving airlines the motivation and justification to adapt their business model and adopt cross-sell strategies. [7]

Australia's Qantas Airways are one of the pioneers in integrating cross-sell technology into their online sales strategy, implementing their first third-party supplier four years ago. A user of Amadeus' e-commerce solutions since 2000, Qantas' website development is supported by Amadeus from a dedicated competency centre in Amadeus' Product and Development base in Sophia Antipolis, France. [7]

Back in 2007, prior to the crippling impact the global financial crisis would have on air travel, Qantas was already looking for new ways to expand their e-commerce offering, so they would stay competitive and increase their revenue. With the travel industry 'buzzing' about the ancillary phenomenon, Qantas' General Manager Direct Channels, John Lonergan, recognised there had to be huge potential in securing untapped revenues from the sale of third-party travel services, especially from their substantial client base of high-yield, frequent flyers. In addition, the positive knock-on effect of enhanced customer satisfaction levels and improved web retention was a major attraction. [7] However, starting to do something new always has a price. In Qantas case it was all about solving few very important things:

- Identify new revenue sources to complement existing online flight booking engine.
- Deliver a full-service travel site, to offer customers an improved and more complete shopping experience.

- Source an IT partner that could offer fully integrated, flexible and multi-touch point technology, without compromising flight sales.
- Have the right technology in place to be able to focus on the commercial relationship with 3rd party providers.

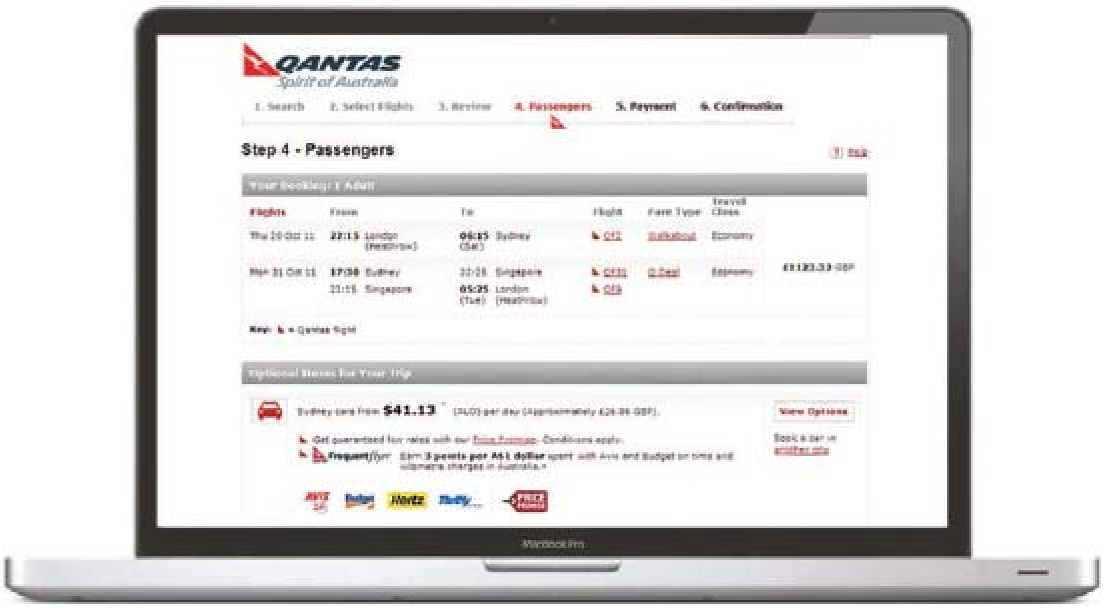
Whilst there were many service providers in the market who offer integration of third-party content into airline booking engines, Qantas weren't looking for a middle man technically or commercially. It was more about improving what they were using already.

Given the existing commercial relationship that Qantas had with QBE Travel Insurance via the Qantas Frequent Flyer programme, it made sense to first incorporate travel insurance into the online sales path. Previously, Qantas had offered QBE services via a separate transaction, however it was considered that full integration within the booking engine would help maximise sales opportunities, and ensure more customers were travelling with valid insurance policies.

Although QBE was not an existing insurance provider in Amadeus, at Qantas' request QBE and Amadeus worked together to make QBE content accessible via the Amadeus GDS. The subsequent integration into the Qantas booking engine proved to be a speedy and straightforward process for the airline. Furthermore, Amadeus provided a valuable point of sale for QBE, who additionally benefited from the subsequent growth in Qantas insurance policy sales.

Customers are offered a selection of insurance policies during various stages in the booking process, either whilst they are buying their air ticket in the purchasing page, or when retrieving their booking in the servicing page. Passenger data is pre-filled from existing passenger name record (PNR) information, and policies are automatically priced, facilitating a quick and easy purchasing process. The success of the insurance implementation was immediate, with Qantas seeing a return on investment one month after launch, and conversion rates of up to 8% on air tickets purchased, compared to 4.6% average achieved by all airlines.

Following on from the successful integration of travel insurance, in 2008 Qantas implemented Amadeus Cars in two formats: as a stand-alone booking solution (where a car rental booking is made independently from the air booking) on the airline's home page, and as a cross-sell solution (where the car booking is added to the same passenger record as the air booking) in the air confirmation page [Figure 14]. There are over 30+ car providers that are seamlessly connected to the Amadeus system including Qantas' four preferred ones that use a 'payment on pick-up' model. As a result, it was a very straightforward process for Amadeus to customise Qantas' website, activate their preferred car suppliers and negotiated rates, in order for Qantas to begin selling car rental to their customers. [7]


1. Search 2. Select Flights 3. Review **4. Passengers** **5. Payment** **6. Confirmation**

Step 4 - Passengers

? Help

Your Booking: 1 Adult

Flights	From		To		Flight	Fare Type	Travel Class	
Thu 20 Oct 11	**22:15**	London (Heathrow)	**06:15** (Sat)	Sydney	QF2	Saleabout	Economy	
Mon 31 Oct 11	**17:30**	Sydney	22:25	Singapore	QF31	Q Deal	Economy	**£1123.33** GBP
	23:15	Singapore	**05:25** (Tue)	London (Heathrow)	QF9			

Key: = Qantas flight

Optional Items for Your Trip

Sydney cars from **$41.13** * (AUD) per day (Approximately £26.86 GBP).

View Options

Get guaranteed low rates with our Price Promise. Conditions apply.

Frequent flyer Earn **3 points per A$1 dollar** spent with Avis and Budget on time and kilometre charges in Australia.*

Book a car in another city

Figure 14. Option to choose a car. [7]

In June 2010, Qantas became the launch partner for Amadeus Shopping Basket. With this new solution, cars can now be booked on the air passenger page, and customers can add or remove car and insurance bookings to or from their unique shopping cart. Upon completion of the booked trip, users can see the total trip amount and can make the full payment in one transaction. With dynamically-priced cross-sell teasers, and a faster, more user-friendly purchasing process, the shopping basket solution contributed to an impressive increase in Qantas' ancillary revenues of approximately 400% in cars, and 100% in travel insurance.

By leveraging technology, in a very short time-frame, Qantas has been able to diversify their online sales strategy and include third party travel services without compromising their core business of flight sales. [7] This shows another huge benefit of GDS – ease of adapting new features and expanding services.

3.4 RESULTS

After doing analysis on what is GDS and how it works, I made an assumption on pros and cons. The good things are centralized system, no need to interact with airlines and airports directly, structured data which can be parsed in a format you need. The bad thing all GDSes have their own data format and usually it is not so easy to parse it as systems were made for human reading instead of machine reading.

Another huge plus was uncovered in Case Study of Qantas 3rd party services. This showed that GDS can not only be used for flight data, but also using flight data generate new services and help to build up new products. This gives an opportunity to offer more for customers and increase the revenue at the same time.

4. Mobile application integration analysis

4.1 INTRODUCTION

After current 15below platform and its key component GDS analysis are done it is time to look further to creating 15below mobile app. In this chapter the main thing is to prepare the mobile app integration into current 15below platform scenario. From previous chapters I already know that of the list of 15below current platform modules, I want to use the following: Ticketing, Disruption and Survey. In addition to this it is important to have a look into PASNGR technical side, what would be API in particular and tools to be used for creating the app concept.

4.2 REQUIREMENTS OVERVIEW

I already mentioned in the introduction the modules to be integrated with the mobile app. Now it is the time to list requirements in detail.

Ticketing

Ticket should be the key component of the app. The user should be able to import his ticket into his app. The app must be a multitenant app allowing storing flights from different carriers in one place. This is very important, because lots of airlines already have their mobile apps allowing storing tickets. The main problem with single tenant app is the need to have an app per airline. If you are a frequent traveler and often flying a different airline, having lots of apps may not only occupy loads of internal device memory, but also to be confusing as your tickets will be all over the places. You can’t call it user friendly. Furthermore, in order to use airline app, you frequently must create an account on their website. The problem with this is not only remembering a list of login credentials, but the registration might be a problem for those less computer literate. So in my app I must avoid registrations as well.

Tickets must be downloaded from PASNGR using API and stored into device internal memory, for offline access. In addition to this, on the main screen there should be a list of tickets in chronological order (the soonest flight first). Every flight in a list must have a short description including carrier, departure and arrival airports, departure date and time. There must be a way of removing and adding new itineraries to the list. On itinerary click, the ticket must open.

The ticket itself must have a carrier logo, to make it more branded, booking reference (PNR), flight number, departure and arrival airport, departure and arrival date and time and most importantly – barcode. Barcode must be in an airline preferred format, so it would be possible to scan it at the departure gate.

Disruption

Mobile device and communication are two things tied together. When there is a disruption, the passenger must be informed as soon as possible. The app must receive schedule changes automatically and must warn the owner even in lock screen. Stay connected while a disruption happens is very important. Furthermore all the flight details should be updated when notification is hit.

If a flight details were changed it must be reflected in itineraries list view, so the user could easily see that there is something new about this flight and he would open it to see what is inside. If schedule was changed, user must be able to confirm his new flight in app. Within the flight details there must be a button which would allow for passenger to confirm his flight and the choice must be fed back into PASNGR. After accepting the schedule change, the itinerary must look as any other itinerary.

Survey

There is only one reliable way to improve services – by listening to your customers. Survey is the most common way of finding out what do customers think of your services. Only one thing, the usual Questions and Answers approach will not work on mobile devices as there are people who do not like to type using on screen keyboard. These customers will skip the survey even if they will have something to say. Rating 1-5 would work perfectly in this situation, having in mind mobile device screen size. Having the right amount of questions customer could fill it in no time.

Questions must be retrieved from PASNGR. They might be different per flight or per customer. Results should be sent back to PASNGR.

Supported platforms

There is a number of mobile OS available on the market. However when launching an app, obviously the main target is offer the product for as big part of market as possible. Looking at the usage statics [Figure 15], we can clearly see the domination of iOS and Android.

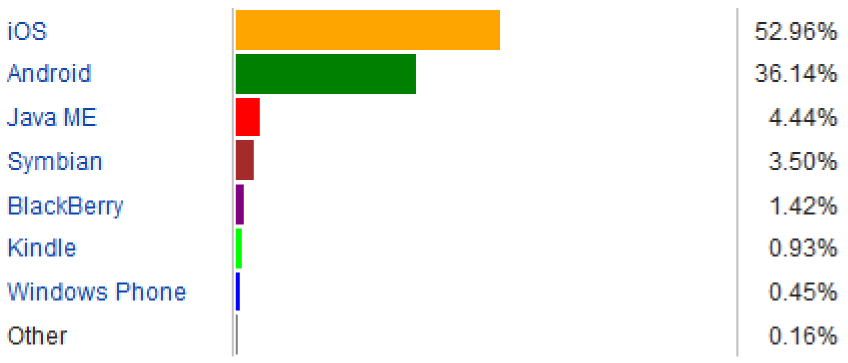


Figure 15. Mobile operating system browsing statistics on Net Applications (February 2014) [8]

Of course the rest OS developers, who only have a very small part of market, still have millions of users and potential customers, so it would be useful to provide a product for them to use as well. Initially the app should work on iOS and Android devices.

4.3 COMPONENTS OVERVIEW

From the first look, there are only two components involved – PASNGR and mobile device. However 15below app should have an extended functionality. To cover this functionality multiple PASNGR modules are necessary. In addition to this, multiple platforms will be used to develop an app on. A basic model of components is displayed below [Figure 16].

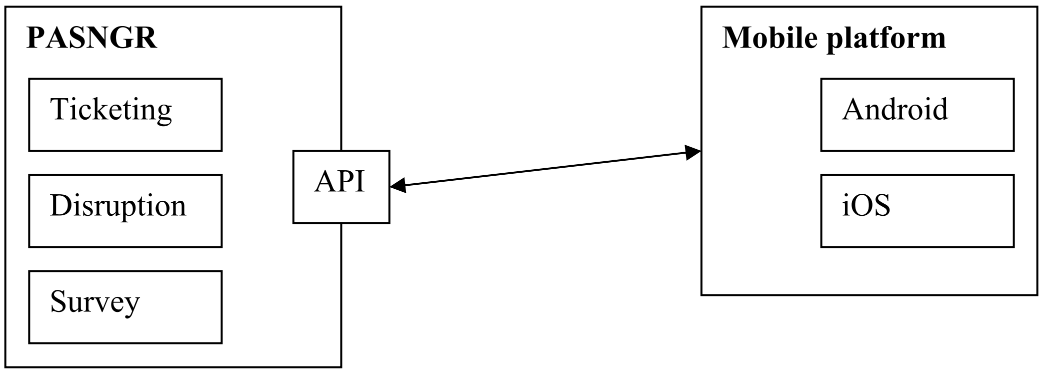


Figure 16. Basic components diagram

As we can see there are two groups of components. There is one essential difference in these groups – mobile platforms are connected directly, while PASNGR modules are accessed over API only.

Looking into each component detail, the best way of doing this would be describing attributes and functions they do so later on, we would be able to build a business model to see what the workflow scenarios would look like.

Starting with Ticketing [Figure 17]. Passenger expects to see all his flight information. This doesn't mean only flight info should be retrieved. As already noted, itinerary will be retrieved using PNR, passenger last name and flight carrier code combination. This obviously suggests that three types of information will be involved in the process. At first we are getting tripInfo for the PNR, then we need associated passengerInfo and his flightInfo. Furthermore, as already mentioned, these details will be retrieved by the app on mobile device, so retrieveTicket function must be available. In addition to this mobile device must be able to send data back to PASNGR, so function updateFlight is also necessary. When app retrieves a trip it must send back device REGID, so PASNGR would know who to send a push notification if necessary. This function will be used by other modules as well so the reasons will be covered later.

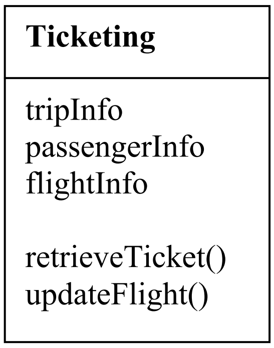


Figure 17. Ticketing module data and functions.

Next component is Disruption [Figure 18]. In case of disruption, updated flight details have to be retrieved, so flightInfo stands here as well. The process consists of three functions. The first one triggerPushNotification is in PASNGR side. If flight is changed, PASNGR sends a push notification to device by REGID associated with the flight. When device receives push notification from PASNGR it automatically should update the flight, so the function retrieveFlightInfo steps in. In case passenger is happy with his flight change it must let airline know, by accepting schedule change. The function acceptScheduleChange is related to updateFlight from Ticketing module, as it updates flight information in the PASNGR with a schedule change accepted flag.

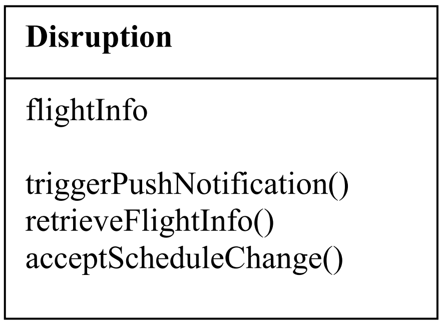


Figure 18. Disruption module data and functions.

The last component in PASNGR group is Survey Module [Figure 19]. Survey only contains questions, so this is the only data it uses. It should retrieve questions using retrieveQuestions function. Obviously after filling in all the answers, they should be sent back to PASNGR. This is where submitAnswers function is used. This function is related to updateFlight function as survey is per flight and after filling in the answers, flight must be updated with survey completed flag, so the passenger won't be asked to answer it again.

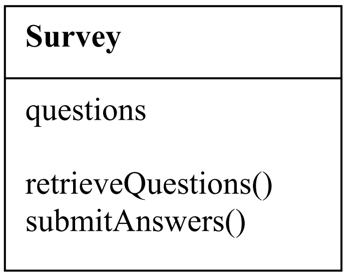


Figure 19. Survey data and functions.

It is worth to mention PASNGR API. It is multitenant API, so any flight could be retrieved using the same API endpoint, what makes easier to implement API access. Also PASNGR API gives JSON object.

Both mobile platforms must have exactly the same functionality so it is not worth to specify data and functions for each of them. It is enough to specify only one, for Mobile platform in general [Figure 20].

Mobile platform

tripInfo
passengerInfo
flightInfo
questions
answers
 	
displayAddItineraryForm()
storeItinerary()
displayIntinerariesList()
displayItineraryInfo()
displaySurveyQuestions()
receivePushNotification()
updateFlightDetails()
acceptScheduleChange()
removeItinerary()

Figure 20. Mobile platform data and functions.

Obviously Mobile platform uses the data provided by different PASNGR modules and interacts with their functions, but is done using internal functionality.

It all begins with displayAddItineraryForm. On submit it calls retrieveTicket to retrieve all the details and then storeItinerary places them in internal device storage. When some itineraries are stored, displayItineraryList is able to display them in a list. On opening any of them, displayItineraryList steps in. Depending on scenario accept schedule change or survey buttons might appear. They would call functions acceptScheduleChange and displaySurveyQuestions. Function acceptScheduleChange would trigger Disruption function with the same name. Function displaySurveyQuestions would display survey questions available for that flight. After filling the survey submitAnswers function from Survey module would be triggered with the answers.

In case of schedule change Disruption module would send a push notification and receivePushNotification function would catch it and use updateFlightDetails function to update flight in device memory.

After flight is expired, or no longer need to be stored, function removeItinerary to be used to removed the flight.

4.4 TOOLS

After requirements and components analysis is done it is quite clear on what has to be implemented. Basically the only one decision is left – to choose the right tools for the implementation. If wrong tools are chosen, the solution might be critical to the whole product and even affect other 15below products. For example if the product requires lots of support, then it will be expensive to maintain. Furthermore if none of current 15below employees are used to the chosen tools, then it would require hiring new employees what also would be costly. Not going into finances as this is not a job of an Analyst Developer, let's do a consideration of available tools and choose the best for the current 15below environment.

Platforms

As already discovered, most of mobile devices either Android or iOS platform. Neither of them is used in 15below at the moment. The most commonly programming language used in 15below is C# for backend and HTML, JavaScript and CSS for frontend. So what are these two platforms?

Android is an operating system based on the Linux kernel, and designed primarily for touch screen mobile devices such as smart phones and tablet computers [Figure 21] [9]. Android is an OS developed by Google.

Figure 21. Android 4.4.2 home screen [9]

From developers’ point of view Androids’ core is written in C, while for frontend C++ and JAVA are used. These programming languages have similarities to C#, but they are not the same and a difference is big enough to affect a quality of possible product.

iOS (previously iPhone OS) is a mobile operating system developed by Apple Inc. and distributed exclusively for Apple hardware [Figure 22] [10].

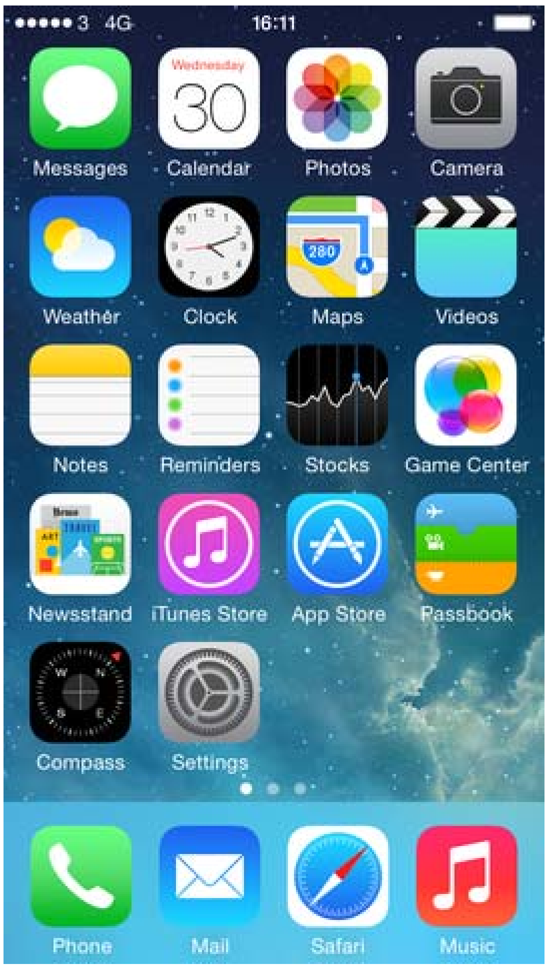


Figure 22. The iPhone home screen in iOS 7.1 [10]

This mobile OS uses C, C++, Objective-C. There are even less Objective-C developers than those who develop in JAVA, so it would be even more expensive to hire one as these languages are also quite different form what 15below is using. Furthermore software could be developed only using Apple software and using only Apple hardware. Also iOS developer license is necessary for developer, which is also paid. This means developing the app specifically for iOS would be very expensive.

Most important thing is we are looking at cross platform solution in order to attract as many users as possible. This would lead to double development as Android app wouldn't work on iOS device and iOS app wouldn't work on Android device.

Looks like the idea of solution which would make apps work on all platforms would be worth a million. Apparently it is too late, there are already cross platform solutions in the market. According to Developer Economics Q1 2013 report (published in November 2013) top 5 cross-platform tools (CPTs) are PhoneGap, Appcelerator, Adobe AIR, Sencha and Qt [Figure 23].

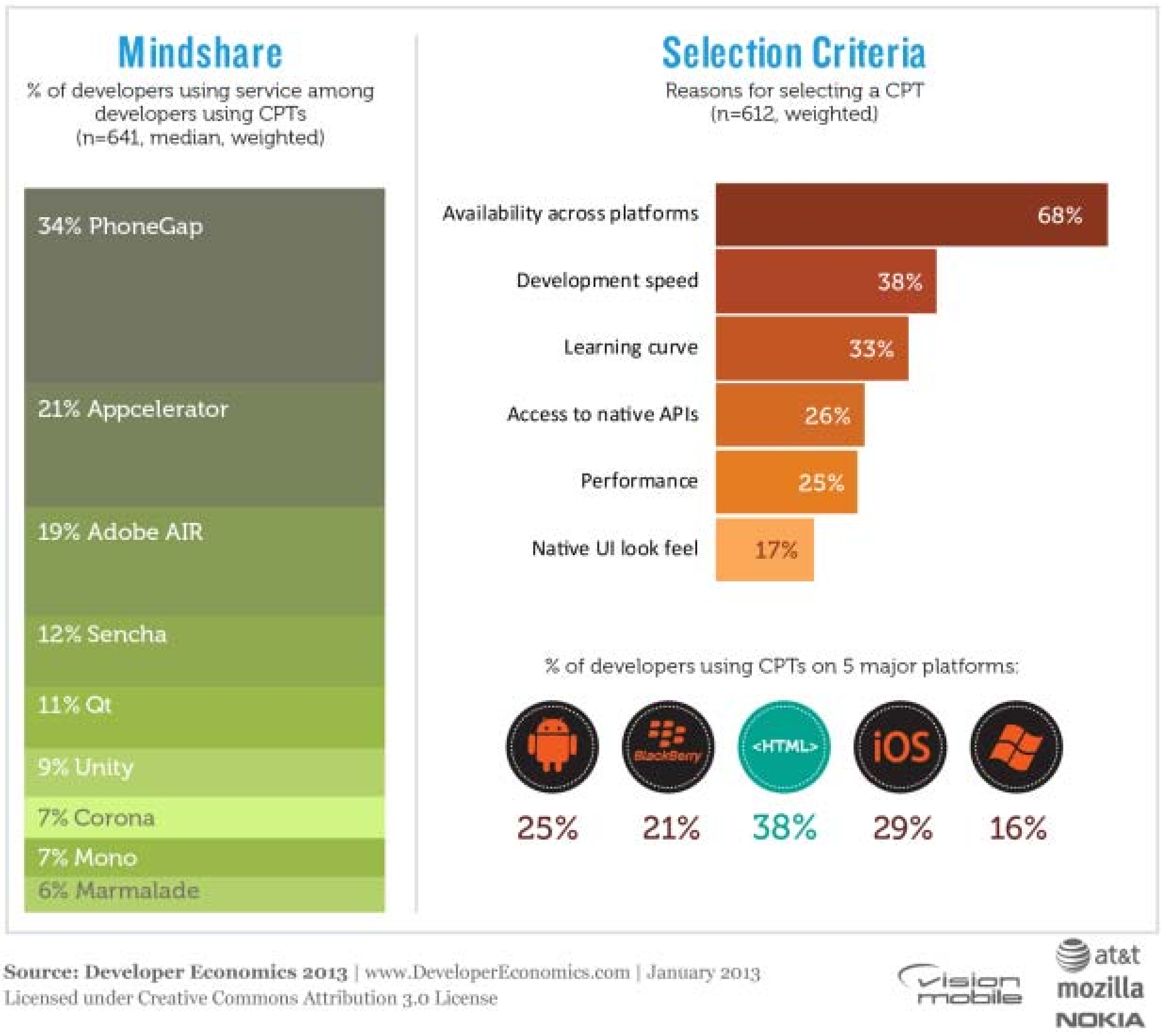


Figure 23. Cross-platform tools. [11]

From the chart above, we can see the domination of PhoneGap. Also there are noted reasons for choosing CPTs. Most of users choose CPTs rather than native solutions like a separate solution per platform, just because availability across platforms and development speed. These two things are very dependant one to another. It is very obvious that an app could be published much faster preparing only one solution for all the platforms rather than a solution per platform.

Of course there are cons in choosing to use CPTs. In my opinion, which matches the chart, the biggest problem is low performance level. For example if you want to make an app which would have lots of internal processes or high quality graphics like 3D games, CPTs would not be an option as in this case the best would be to use tools recommended by platform developers.

There is one more important thing in the chart – HTML. Usually UI side is developed in HTML, which is probably one of the best known computer languages in the world.

The next question would be which cross-platform tool to choose and this would require taking a deeper analysis on each of them.

Apache Cordova (known by many as PhoneGap) holds the top slot in developer mindshare. Cordova/PhoneGap developers write their mobile applications using HTML, JavaScript and CSS. These assets run in a WebView inside a native application container on the target platform. It is, conceptually, a web application packaged within a native application container where your JavaScript has access to device-level APIs that normal web applications would not. [11]

The good things about PhoneGap, that users are developing in JavaScript, HTML and CSS. Furthermore it is compiled and installed like a native app and is modular – easy to add or remove modules, giving an access to various device features. Also it is absolutely free to use. The downside is its performance. If you are developing a large app, you must be a good web developer, so you could optimize it.

Appcelerator's Titanium provides a unified (across devices) JavaScript API, coupled with native-platform-specific features. Developers write JavaScript and utilize a UI abstraction (the Alloy MVC framework) that results in the use of native UI components, greatly aiding UI performance compared to other hybrid options. [11]

The pros of Appcelerator are usage of JavaScript to develop and use of native UI components what will improve the performance. However native UI component is also a con as developer must be trained to use them for each platform.

Adobe AIR is "a cross-operating-system runtime that lets developers combine HTML, JavaScript, Adobe Flash® and Flex technologies, and ActionScript® to deploy rich Internet applications (RIAs) on a broad range of devices including desktop computers, netbooks, tablets, smartphones, and TVs." The problem with that description is that you cannot use HTML & JavaScript to write Adobe AIR applications for mobile applications (Flash/ActionScript skills need only apply).

Adobe AIR is good for better animations and those users who have experience working with Adobe Flash using ActionScript. However the fact that Adobe purchased Nitobi (and the rights to the PhoneGap name), clearly signaling to many that AIR may not be a long term strategy for mobile development.

Sencha Touch is an HTML5 mobile application framework for building web applications that look and feel like native applications. Apps built with Sencha Touch can be used with Apache Cordova/PhoneGap or Sencha's native packager – either which will package the application in a native container and enable access to select device-level APIs unavailable to traditional web apps. [11]

Sencha has a list of products made for mobile apps development, from Architecture tools to IDE such as Eclipse plug-ins, those should make development easier. However Sencha is compiling apps via PhoneGap if you use PhoneGap plug-ins, so the user would need to know both of those platforms.

Qt ("Cute") is a cross-platform development tool that targets a number of embedded, desktop and mobile platforms. Developers write using "QML", touted as a "CSS & JavaScript like language", and apps are backed with an extensive set of C++ libraries, and utilize graphics/UI components written in C++. [11]

The main advantage of Qt it has it own IDE tooling (Qt Creator IDE & Qt Designer) appear to be solid development tools, and code profiling is available in QML Profiler. However it is paid software.

All things considered PhoneGap looks like the best solution for our app. The reason behind this is JavaScript, HTML and CSS used for development. These are languages well known for any front end developer. Furthermore any developer in 15below knows how to use these languages and anyone can choose his preferred IDE. This would mean an easy start. Also PASNGR API is based on JSON so JavaScript is a perfect match for parsing data. Of course in analysis was a concern about performance, but 15below mobile app, with required functionality, will be too small so it would have performance issues.

Notifications Delivery

In case of an event like schedule change, the user must be notified. Also notification must include some payload data, which would be used for getting new details. As a solution Android is using Google Cloud Messaging (GCM) service while iOS uses Push Notifications. In theory they are very similar, however they are platform dependant, so there are no cross platform solutions here. The good thing though is PhoneGap supports both of them, so once device receives a message, the app built on PhoneGap can be triggered and access the payload data.

Google Cloud Messaging for Android (GCM) is a service that allows you to send data from your server to your users' Android-powered device, and also to receive messages from devices on the same connection. The GCM service handles all aspects of queueing of messages and delivery to the target Android application running on the target device. GCM is completely free no matter how big your messaging needs are, and there are no quotas. [12]

GCM message is able to transfer up to 4kB of data in JSON format. In order to get the message, the app must have Google API project ID and device registration ID. Device registration ID has to be sent to PASNGR API, as in case of event PASNGR would know which device to send the notification to.

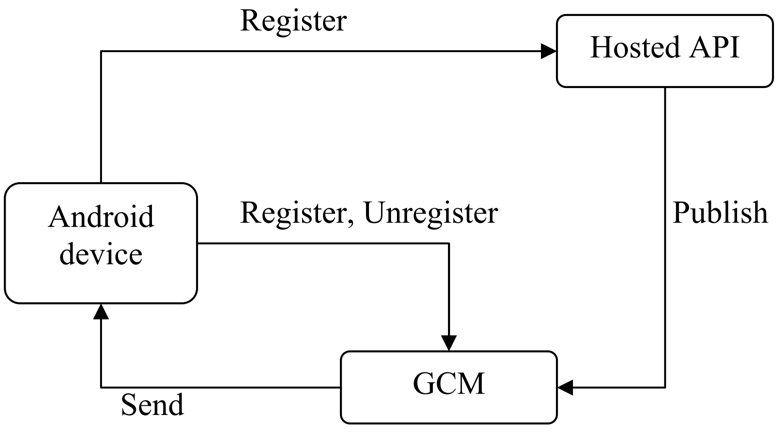


Figure 24. Google Cloud Messaging service workflow.

As per Figure 24 an Android Device must register to Hosted API, in our case PASNGR, and GCM. This has to be done in order for PASNGR and GCM to know to which device send the message. Additionally unregister can be performed with GCM in case of application removal from the device. When the Hosted API has the Registration ID of the device it can publish a message on GCM. GCM having Registration ID of the device and Project ID of the app, can send the notification to the targeted app on the targeted device.

Apple Push Notification service (APNs) propagates push notifications to devices having applications registered to receive those notifications. Each device establishes an accredited and encrypted IP connection with the service and receives notifications over this persistent connection. Providers connect with APNs through a persistent and secure channel while monitoring incoming data intended for their client applications. When new data for an application arrives, the provider prepares and sends a notification through the channel to APNs, which pushes the notification to the target device. [14]

APNs has a more complicated way of registering device and sending notifications, so it is split into two workflows.

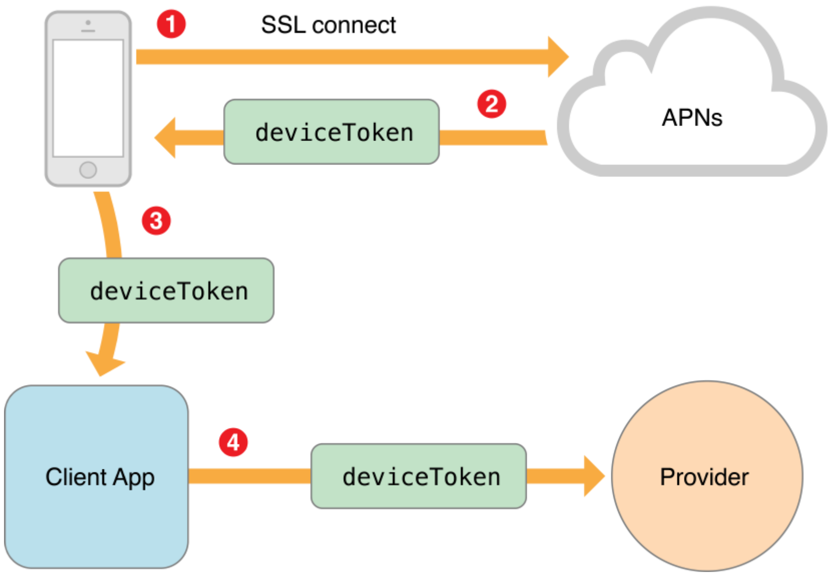


Figure 25. Registering a device on APNs. [14]

As shown in Figure 25, the device at first connects to APNs to get a deviceToken. Then the deviceToken is given for an installed app to use it. The app then will register its token with the provider, what is in our case PASNGR.

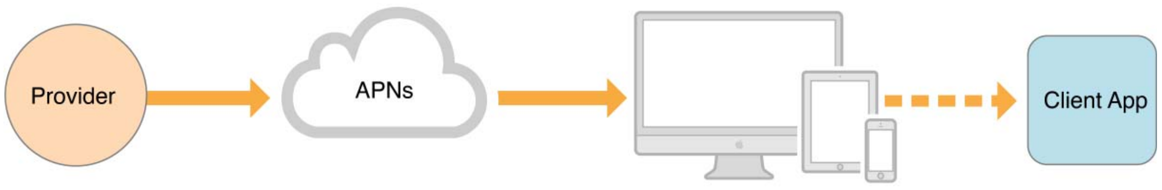


Figure 26. Sending a message through APNs. [14]

Figure 26 represents notification sending to the iOS device. This workflow is quite basic. When the provider has the deviceToken it just publishes the message on APNs using it. Then APNs sends it to the device and device itself to the app. This part is almost the same as used by GCM send.

4.5 BUSINESS PROCESS MODEL

In order to get the better idea how the app should work it is always highly recommended to have a business process model. It would represent the sequence of business processes inside the app. My business process model represents the full functionality of 15below mobile app [Figure 27].

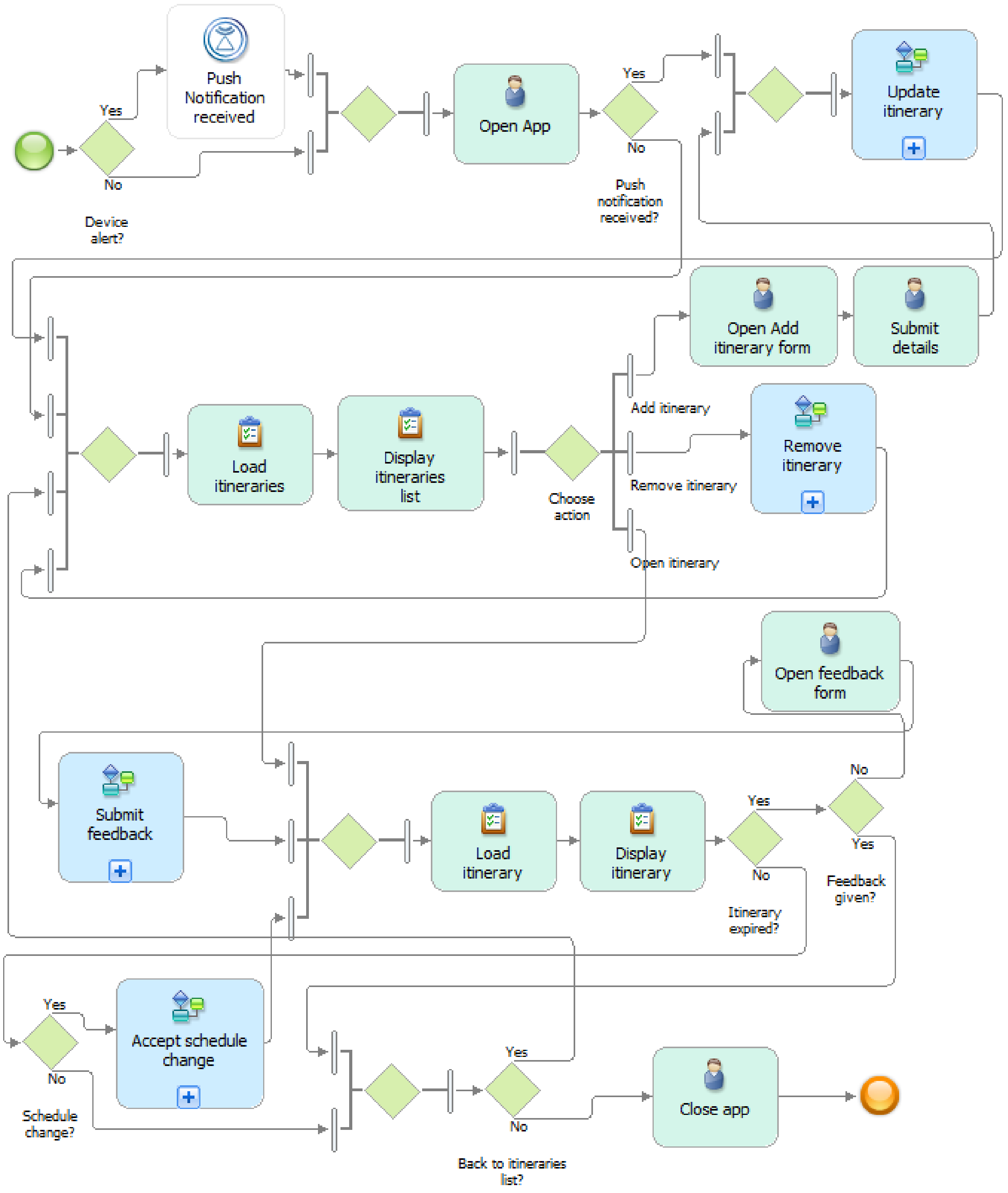


Figure 27. 15below mobile app functionality business process model.

The process begins with user opening the app. There are two ways of doing this. First one in case user is alerted by received push notification and the second one when user wants it. When app is opened, internal decision takes action. If push notification was received, it means there is an update for stored itinerary. In this case internal process Update itinerary [Figure 28] is triggered.

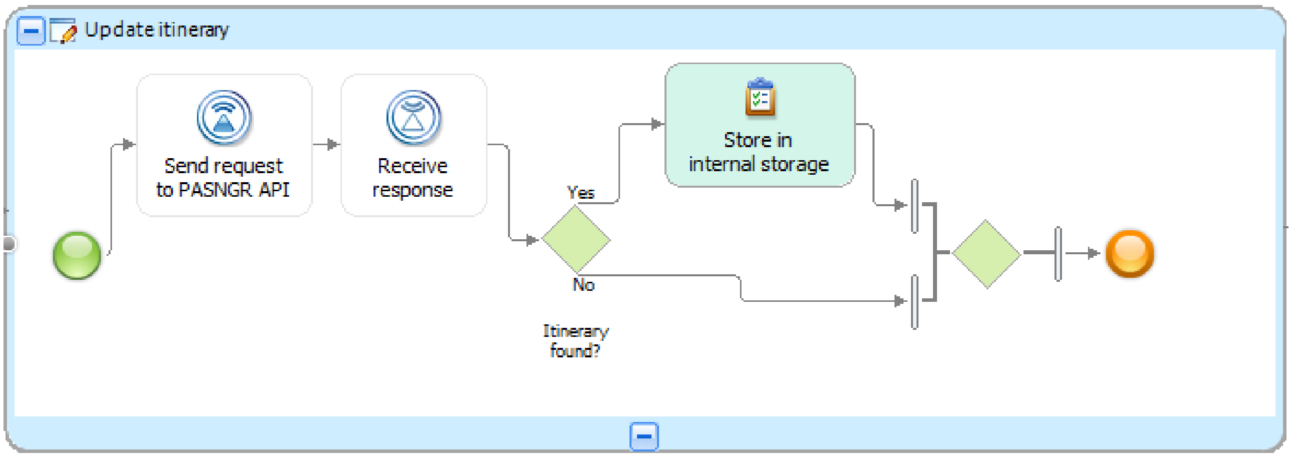


Figure 28. Internal process Update itinerary.

Firstly a request to PASNGR API is sent. Then a response is received. If itinerary was found then the details are stored in the device's internal storage.

Looking back into the main model, we can see that the next step merges few workflows. One of branches contains the flow, when the user opens the app on his choice and not after receiving push notification. Going further we see internal task Load itineraries. This task basically loads the itineraries from the internal storage and then proceeds to the task which does the display of pre-loaded itineraries. Then we can see a list of actions those represents the functionality of current UI view. The user can add, remove or view itineraries of his choice. If add itinerary is chosen Add itinerary form opens. Then the user enters details and workflow proceeds to the internal process I have already analyzed – Update itinerary. If Remove itinerary is chosen, then the workflow proceeds to the internal process called Remove itinerary [Figure 29].

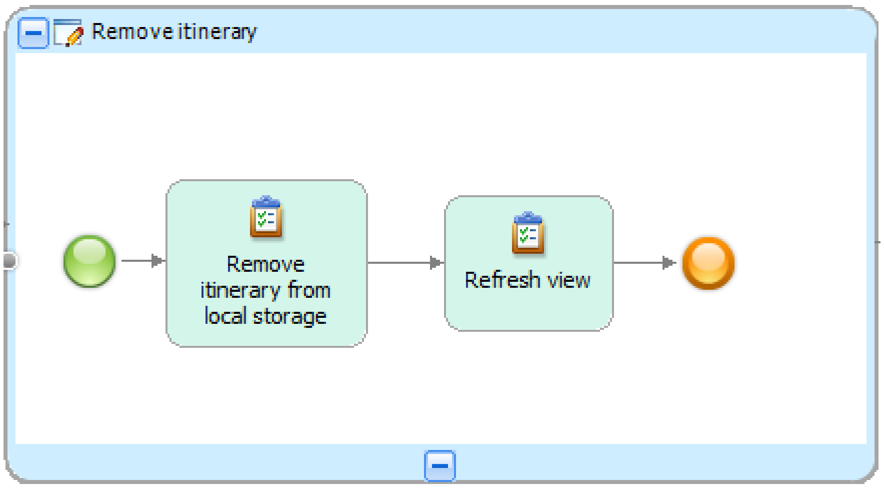


Figure 29. Internal process Remove itinerary.

Internal process Remove itinerary is quite simple. It contains only of two internal tasks. First one does the remove of itinerary from internal storage and the second one refreshes the view of itineraries, so the removed one would be no longer displayed.

In the decisions choice in the main model the third option was Open itinerary. This choice opens a single itinerary view, by loading the details for chosen itinerary and displaying them. In case itinerary is expired, the flight is in past, the user is offered to give a feedback on his flight experience. If user wants to give a feedback, he can open Feedback form. Submitting feedback goes to yet another internal process [Figure 30].

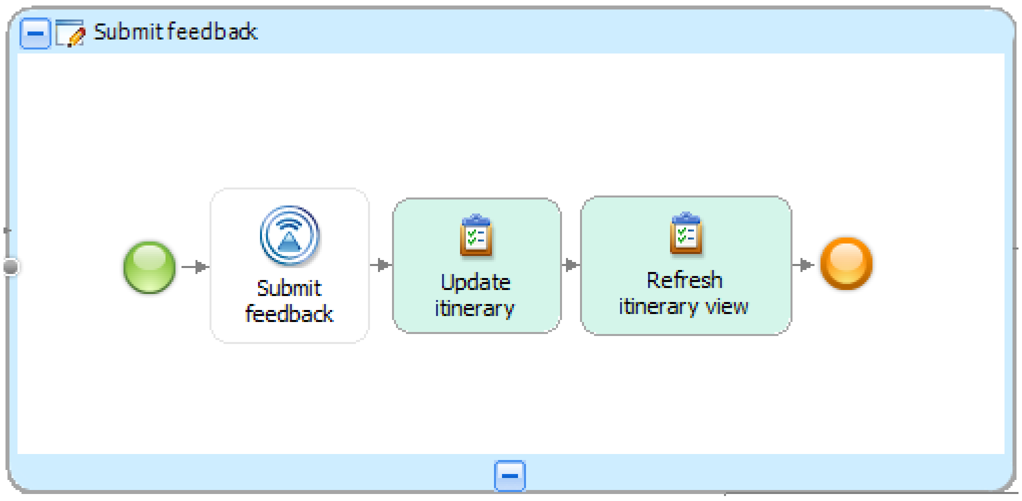


Figure 30. Internal process Submit feedback.

Feedback has to be sent as a request to PASNGR API. Once feedback is sent the flight must be updated in the internal storage, so it won't offer to do this twice. After that itinerary view is refreshed.

Looking back again to the main model, there is an internal decision if itinerary was updated because of schedule change. If yes, then the user must accept his schedule change in order for airline to know, that the passenger is happy with his recommendation. So the workflow proceeds to internal task called Accept schedule change [Figure 31].

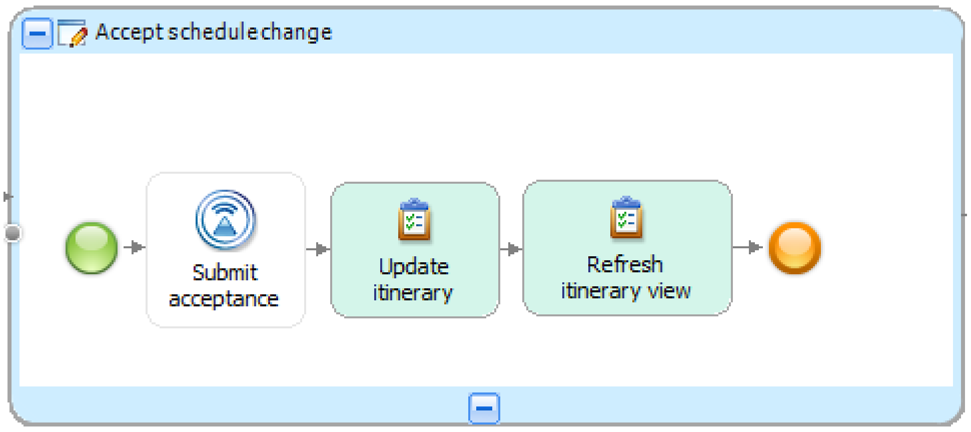


Figure 31. Internal process Accept schedule change.

Firstly the request is made to PASNGR API. Then the itinerary is updated in the internal device memory, in order not to display schedule change notification for the user. After details are updated, the itinerary view is refreshed.

In the main model, finally user can go back to where he started – the list of his itineraries or close the app.

4.6 RESULTS

In the beginning of this chapter, requirements for the 15below mobile app were analyzed. The required functionality includes flight tickets storing with ability to display barcode and store offline, schedule change notifications and acceptances handling, passenger feedback, Android and iOS support. In addition to this those two platforms analysis was done, considering them to be most popular in the mobile apps market. As some required functionality requires messaging, messages delivery processes were overviewed. In the result they were considered to be different per platform. Furthermore for implementing the app Adobe Phonegap was chosen. The reason behind that is programming languages used – JavaScript, HTML5 and CSS. These languages were found as the best choice for current 15below developers to use. Finally, as everything has to be integrated with current 15below platform, a business process model has been modeled to show how the app will work and communicate with PASNGR API.

5. Mobile application prototype

5.1 INTRODUCTION

The final and probably the most important part of this dissertation is creating the concept. There were already done analysis on PASNGR components, GDS, requirements, integration and tools. Now it is the time to join them together in a mobile app that would work on iOS and Android powered devices. In this chapter there will be highlighted tools used to create the app and screenshots covering required functionality.

5.2 SOLUTION

In previous chapter it was found that the best would be to use Adobe Phonegap for creating the app. It is good because all code is written in JavaScript, HTML and CSS. Also for developing in JavaScript you choose your favorite IDE as most of IDEs support JavaScript. Another good reason is – single code base for all platforms. Phonegap also has free plug-ins, those let us easily extend functionality of an app. In my case I am using PushPlugin as my app must support Push Notifications. Using already prepared plug-in for that lets me to avoid extra coding.

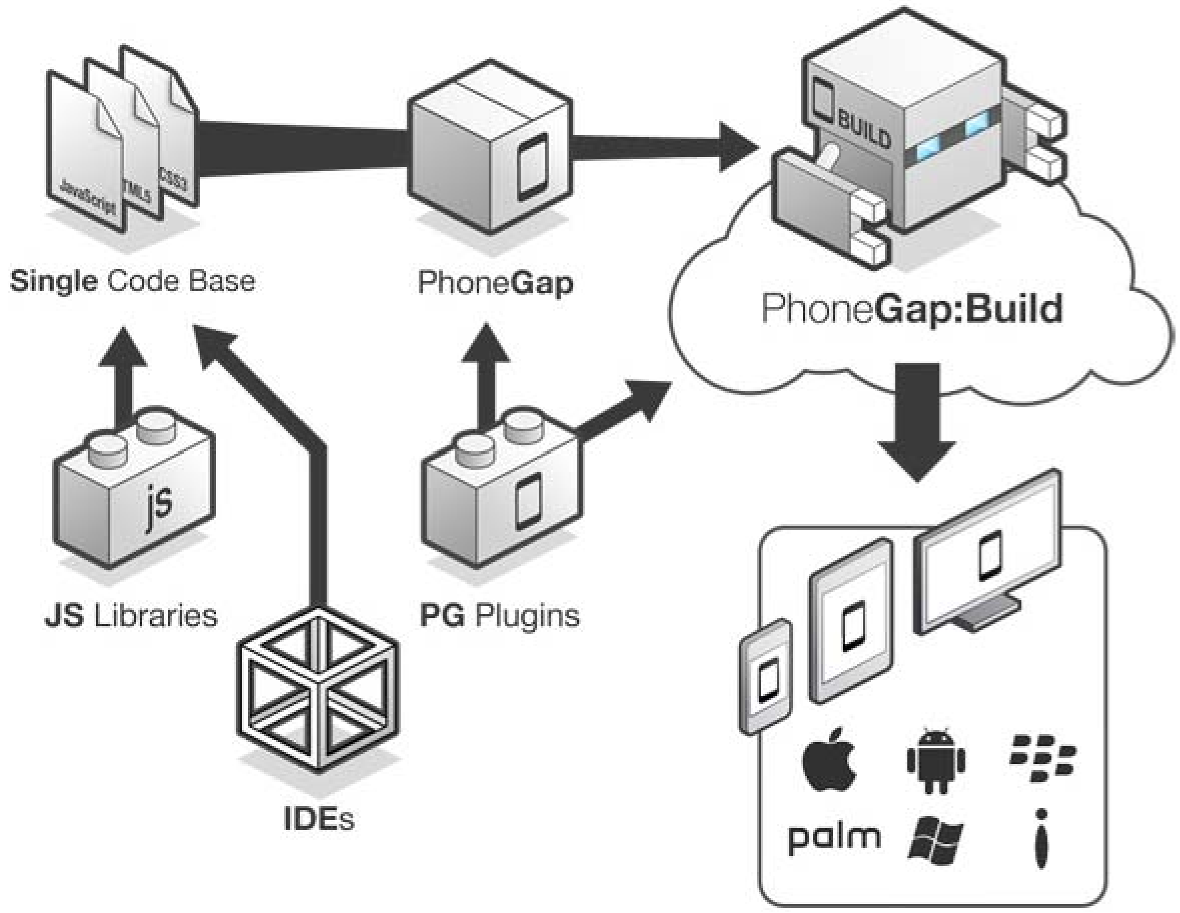


Figure 32. Visualization of Adobe Phonegap [14].

However along with the source code and graphical images it still needs to add some configuration. The good thing there is not much of it and it is very obvious for any developer. When all of this is ready, it needs to be zipped and uploaded to the build.phonegap.com website, which compiles the source code into an app and gives back a QR code, so we could scan it and get the app installed on the preferred device [Figure 33].

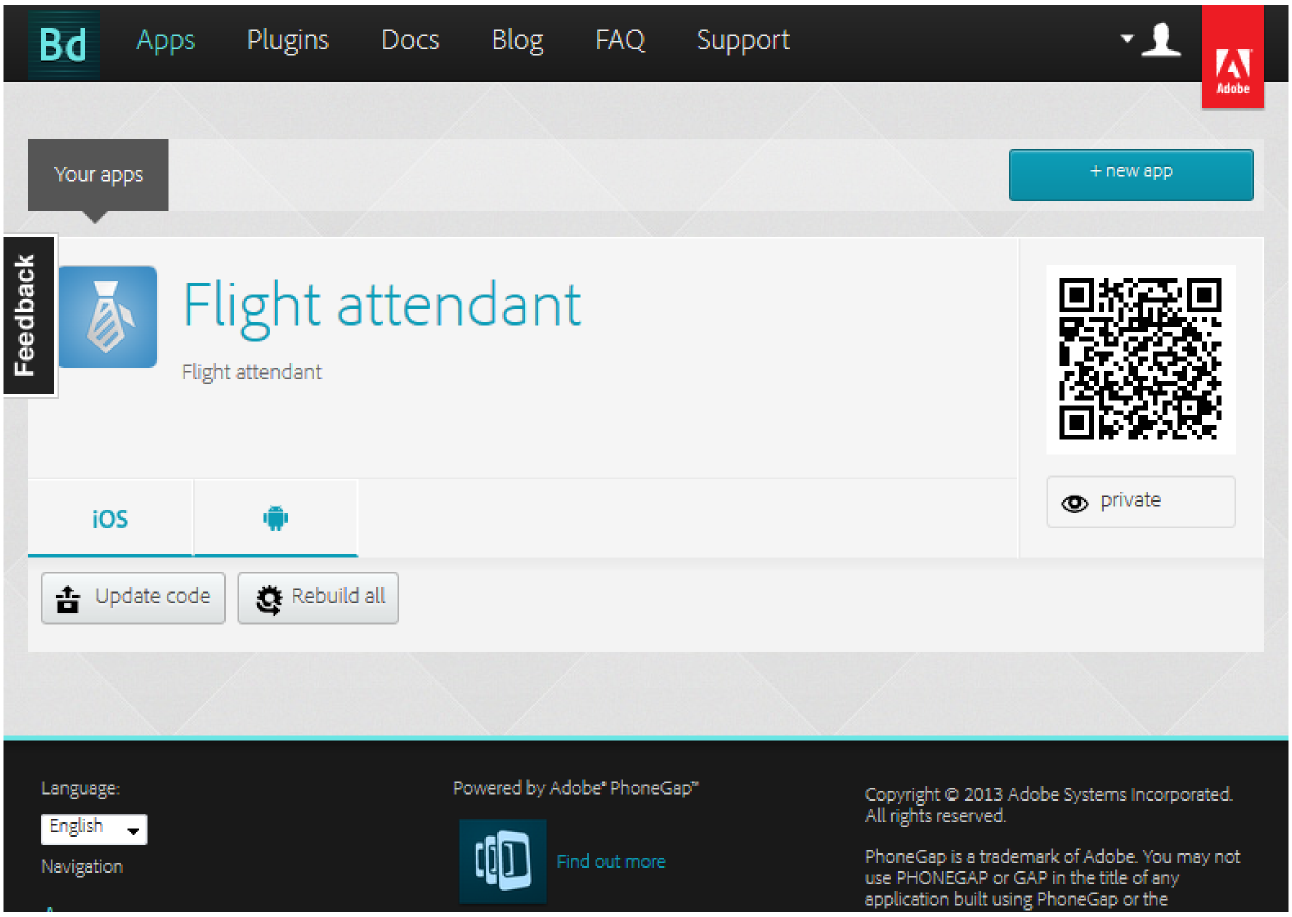

Bd
Apps
Plugins
Docs
Blog
FAQ
Support
Adobe
Your apps
+ new app
Feedback
Flight attendant
Flight attendant
iOS
private
Update code
Rebuild all
Language:
English
Navigation
Powered by Adobe® PhoneGap™
Find out more
Copyright © 2013 Adobe Systems Incorporated. All rights reserved.
PhoneGap is a trademark of Adobe. You may not use PHONEGAP or GAP in the title of any application built using PhoneGap or the

Figure 33. Adobe Phonegap build main screen.

It is now obvious from the figure above, that I named the app as Flight attendant. Also as I mentioned, scanning the given QR code with iOS or Android device installs the app on it [Figure 34].

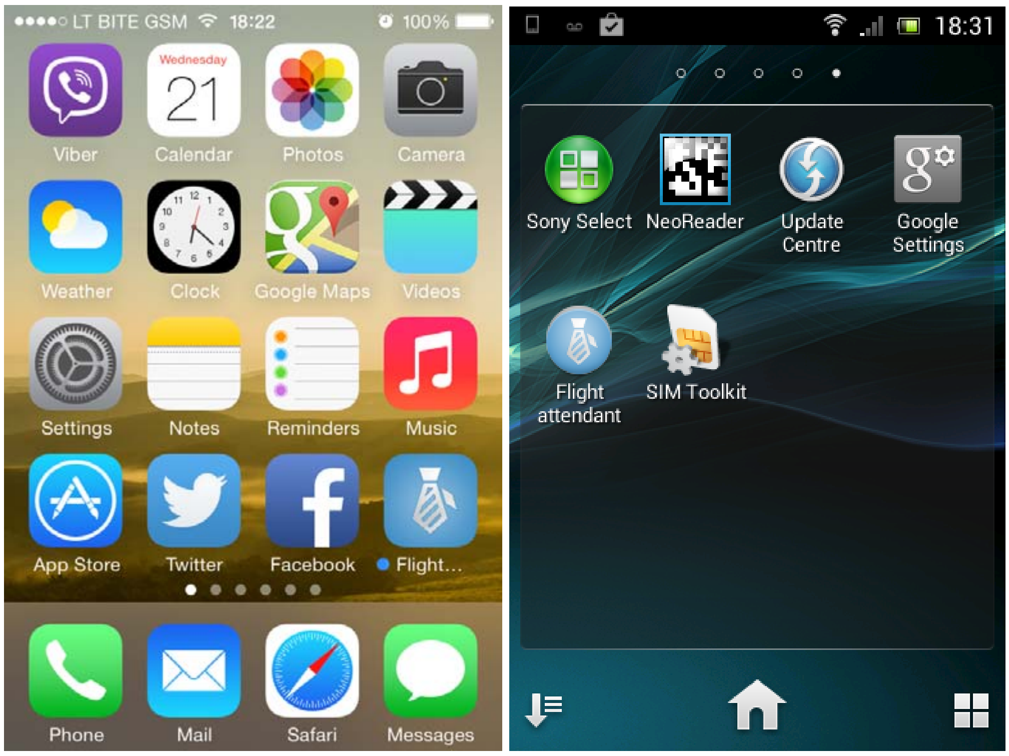


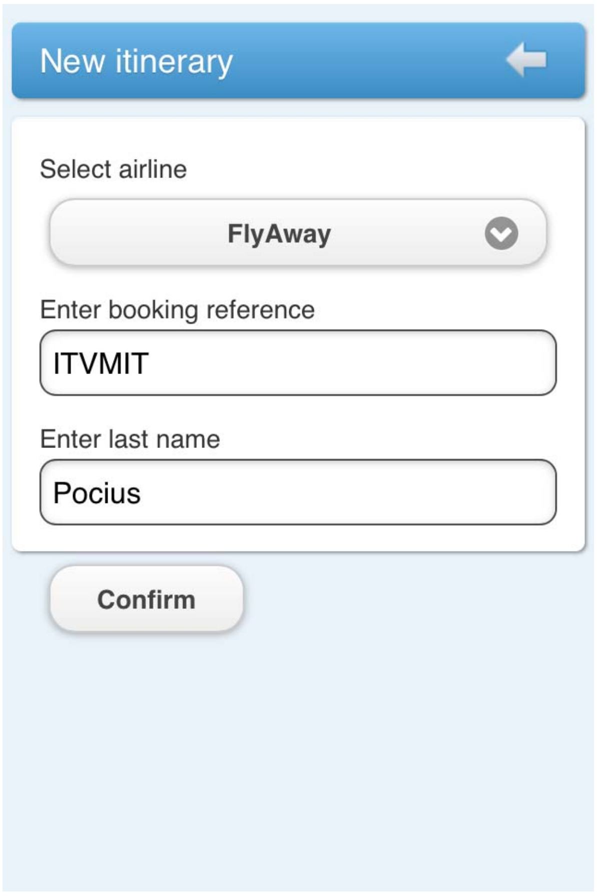
Figure 34. Flight attendant app installed on iOS 7 and Android 4.2 devices.

Once the app is installed, it holds no data and if the app is opened it looks pretty much empty [Figure 35]. The plus sign means "Add itinerary to the list" and the pencil sign means "Edit itineraries list". Obviously if any is saved on the device.

Itineraries

Figure 35. The main app screen.

As I mentioned the plus sign will let user to add an itinerary, so if this button will be hit, add itinerary form will open [Figure 35].



Select airline

FlyAway

Enter booking reference

ITVMIT

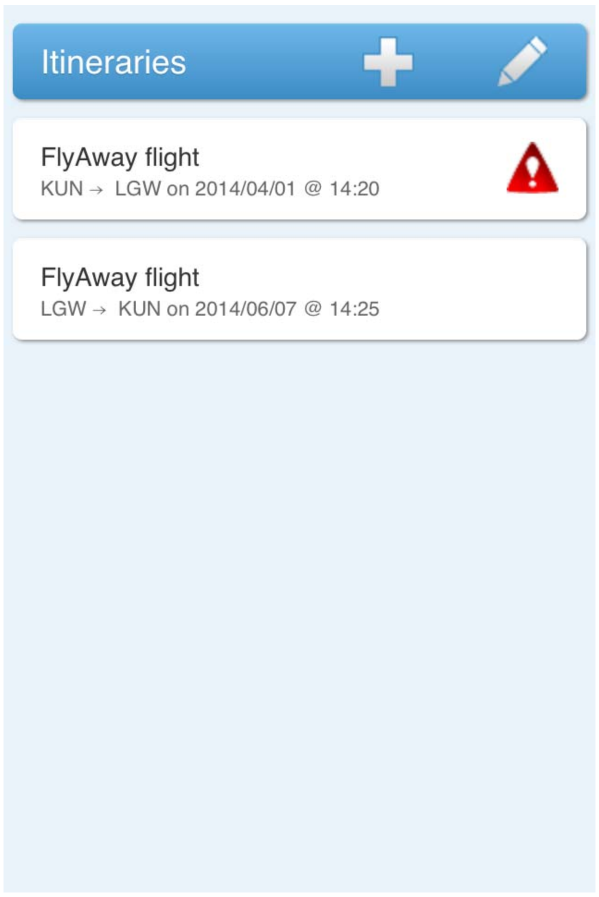
Enter last name

Pocius

Figure 36. Add itinerary form.

While doing analysis I have found that itinerary key must contain Airline code, Booking Reference and lead passenger last name. This is because booking reference is not unique not only per GDS, but even per airline. Furthermore it is very obvious that the rest of two details are not unique either. This is why we need to fill them in.

Once the correct details are filled in and confirm button is hit, there is a web service call made to PASNGR API and JSON response containing itinerary info is returned back to the app. The details are saved internally for ability to use them offline. In addition to this a list of flights those were associated to the itinerary are now visible in the main screen [Figure 37].

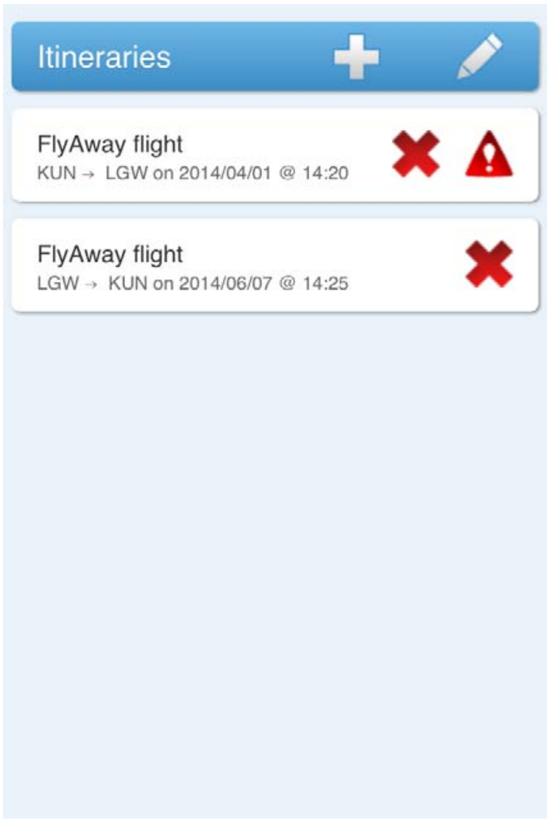

Itineraries

FlyAway flight
KUN → LGW on 2014/04/01 @ 14:20

FlyAway flight
LGW → KUN on 2014/06/07 @ 14:25

Figure 37. List of itineraries.

At this point, if the pencil sign will be hit, the red x will appear on each flight [Figure 38]. This will give an option to remove the flight from the phone in case you no longer need it. For example your flight is expired. Hitting the red x will remove the flight from the device only. It does not do any call to API to remove the flight from server.

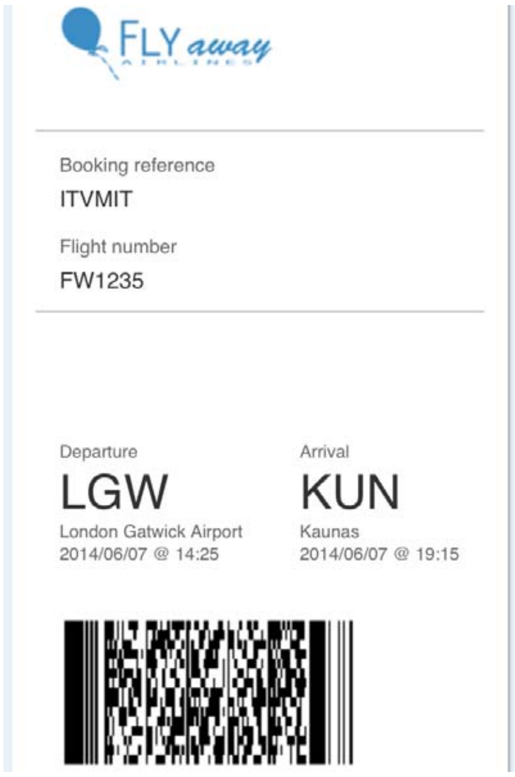


FlyAway flight
KUN → LGW on 2014/04/01 @ 14:20

FlyAway flight
LGW → KUN on 2014/06/07 @ 14:25

Figure 38. Option to remove a flight.

On one of the flights there is an exclamation mark visible. Obviously it means a notice which could be find inside. In order to open the flight user has to hit on it and user will be taken to the view itinerary window. Say we hit the second one for now, which has no message inside, so we can see the flight e-ticket with basic flight information and a barcode which could be scanned at the airport [Figure 39]. The carrier logo is displayed per carrier, to add some branding to the e-ticket.


Booking reference

ITVMIT

Flight number

FW1235

Departure

LGW

London Gatwick Airport

2014/06/07 @ 14:25

Arrival

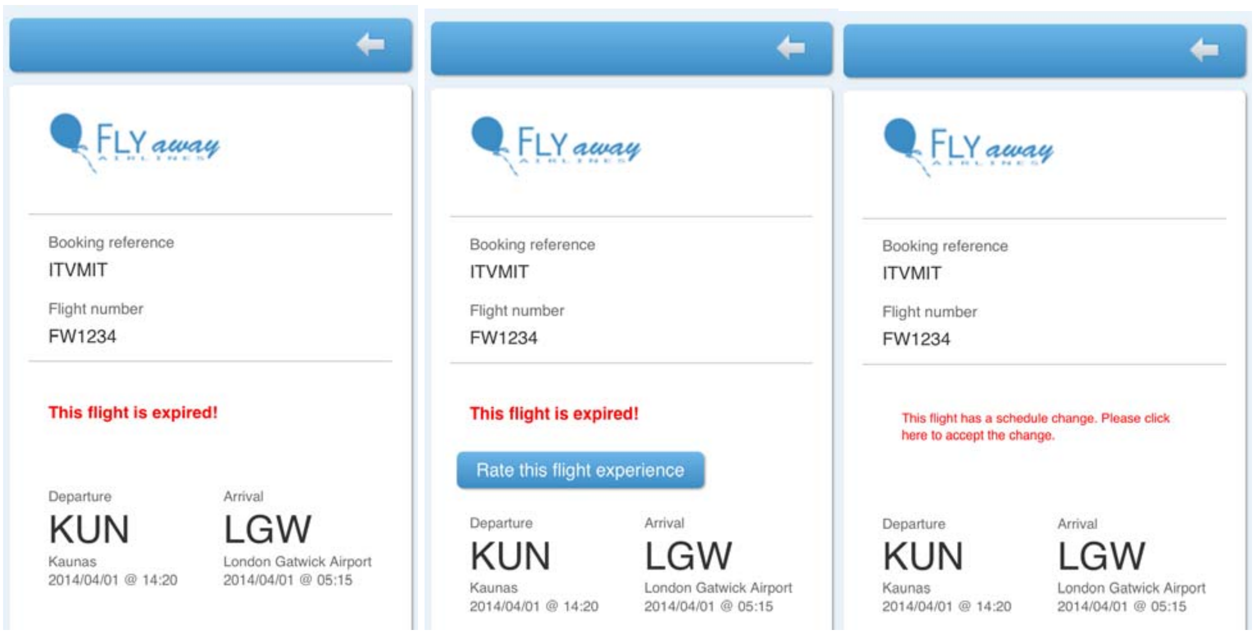
KUN

Kaunas

2014/06/07 @ 19:15

Figure 39. E-ticket stored on Flight attendant mobile app.

If we think about the main screen again, there was a flight with a message. If we open that flight it will display some extra information [Figure 40].


Booking reference

ITVMIT

Flight number

FW1234

This flight is expired!

Departure

KUN

Kaunas

2014/04/01 @ 14:20

Arrival

LGW

London Gatwick Airport

2014/04/01 @ 05:15

Booking reference

ITVMIT

Flight number

FW1234

This flight is expired!

Rate this flight experience

Departure

KUN

Kaunas

2014/04/01 @ 14:20

Arrival

LGW

London Gatwick Airport

2014/04/01 @ 05:15

Booking reference

ITVMIT

Flight number

FW1234

This flight has a schedule change. Please click here to accept the change.

Departure

KUN

Kaunas

2014/04/01 @ 14:20

Arrival

LGW

London Gatwick Airport

2014/04/01 @ 05:15

Figure 40. Different types of notice inside a flight.

As we can see there could be three types of different notices:

- Expired flight notice;
- Expired flight notice with an offer to rate the flight;
- Schedule change notice;

Expired Flight notice appears when the flight was in past. Usually the message appears along with “Rate this flight experience” button. If this button is hit, user is taken to rate flight experience view [Figure 41].

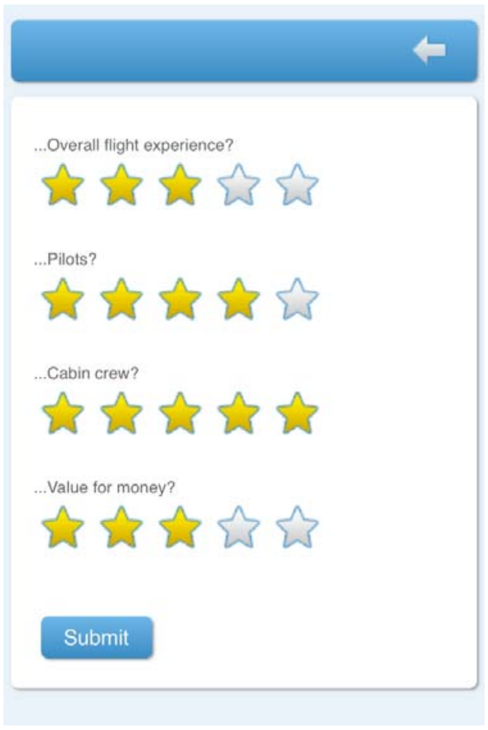
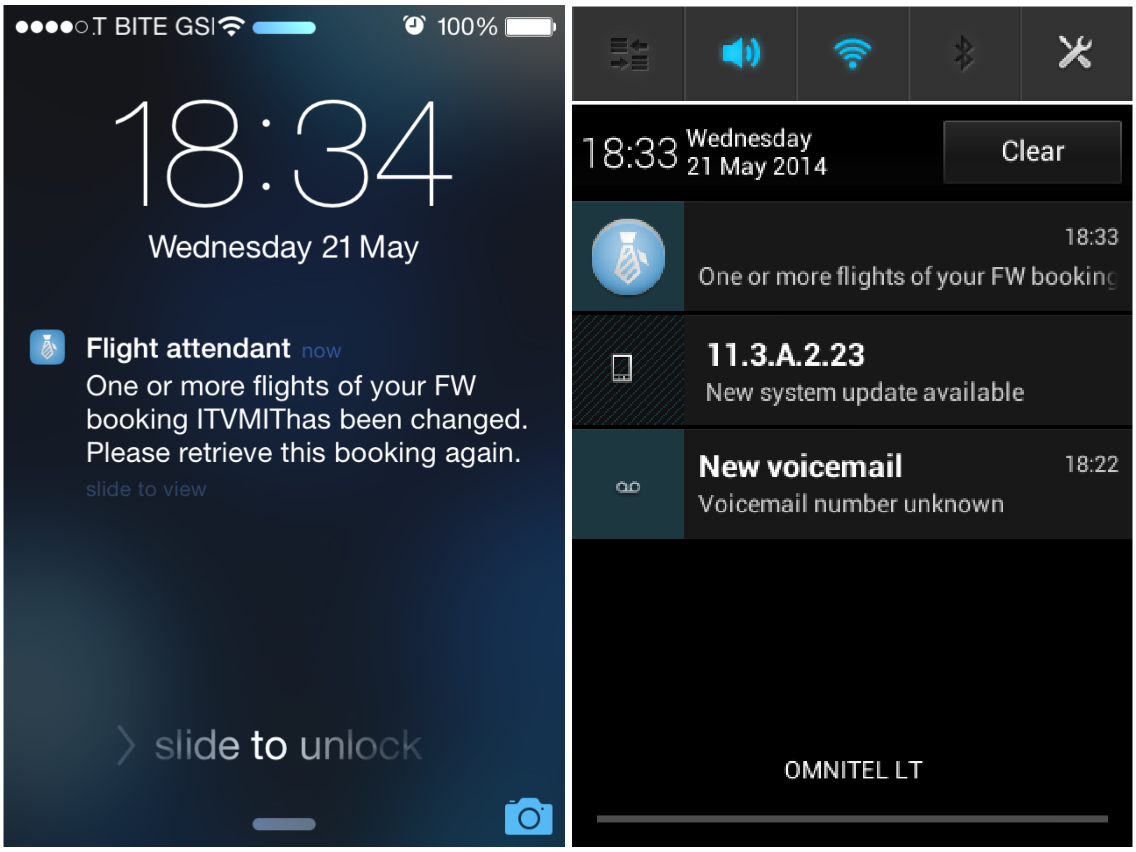


Figure 41. Rating flight experience.

Users might not want to write long reviews using on screen keyboard, so using the star rating for short questions could be very effective way to listen to customers. The questions are created per airline and they are retrieved at the same time itinerary is retrieved. Once all stars are given and submit button is hit, web service call sends the results back to PASNGR API and the blue button disappears form the flight view.

The notification for schedule change appears when there is a schedule change for the flight. User gets notified by a push notification [Figure 42] and the stored flight details are changed.


●●●●○T BITE GSl 100%

18:34

Wednesday 21 May

Flight attendant now
One or more flights of your FW booking ITVMIThas been changed. Please retrieve this booking again.
slide to view

slide to unlock

Clear

18:33 Wednesday 21 May 2014

18:33
One or more flights of your FW booking

11.3.A.2.23
New system update available

New voicemail 18:22
Voicemail number unknown

OMNITEL LT

Figure 42. Schedule change notification received on iOS and Android device.

Hitting the notification for schedule change text (Figure 41) will result a call to PASNGR API to accept schedule change and let an airline to know that passenger is happy about his new flight. Alternatively passenger will know that there is a schedule change and he needs to contact the airline to check other options.

5.3 RESULTS

As a result I have created a mobile app for iOS and Android devices which is able to retrieve itineraries and store them for offline use. The itineraries contain barcode, to be able to scan it at airport and use as an e-ticket. In addition to this the app supports schedule change notification via push notifications services and rating post flight experience. These are all requirements listed in previous chapters and they now are fully implemented. The most importantly the app is integrated with current 15below platform PASNGR. At the end we have an app which requires no registrations, meaning no remembering of passwords and with simple user interface it is relatively easy to use even for those non computer literate.

6. Final results

As a first thing in the project 15below platform PASNGR analysis was done to decide which of modules could be replicated in 15below mobile application. There was made a decision as a core function to use Ticketing, so the user could store his tickets on his mobile device and have them wherever he goes. In addition to this Disruption module could be integrated with ticket and provide any disruption information to the traveler promptly. Schedule change acceptance should go as a part of functionality, so the user could accept new flight with one button click. Also Survey module could be integrated as a part of post trip customer satisfaction feedback.

To decide what data and workflows stand behind PASNGR analysis on GDS was done assuming pros and cons. The good things are centralized system, no need to interact with airlines and airports directly, structured data which can be parsed in a format you need. The bad thing all GDSes have their own data format and usually it is not so easy to parse it as systems were made for human reading instead of machine reading.

Another huge plus was uncovered in Case Study of Qantas 3rd party services. This showed that GDS can not only be used for flight data, but also using flight data generate new services and help to build up new products. This gives an opportunity to offer more for customers and increase the revenue at the same time.

For implementing the app Adobe Phonegap was chosen. The reason behind that is programming languages used – JavaScript, HTML5 and CSS. Also this means a single code base for all targeted platforms. These languages were found as the best choice for current 15below developers to use, knowing the skills range. Finally, as everything has to be integrated with current 15below platform, a business process model has been modeled to show how the app will work and communicate with PASNGR API.

As a final result I have created a mobile app for iOS and Android devices which is able to retrieve itineraries and store them for offline use. The itineraries contain barcode, to be able to scan it at airport and use as an e-ticket. In addition to this the app supports schedule change notification via push notifications services and rating post flight experience. These are all requirements listed in previous chapters and they now are fully implemented. The most importantly the app is integrated with current 15below platform PASNGR. At the end we have an app which requires no registrations, meaning no remembering of passwords and with simple user interface it is relatively easy to use even for those non computer literate.

Bibliographical entries

1. *15below.com* [interactive], [viewed on 10 May 2014]. Online access: <www.15below.com>
2. 15below ltd, *FACT SHEETS - One Pagers - Content for ROOST v1 (6 Jan 2014)* [DOC]. 2014. 22p.
3. Amadeus, *Passengers first: Re-thinking irregular operations* [PDF]. 2013. 26p.
4. Juan Salmoral Franco, *Notification of Reservation Record Simultaneous Changes, Final Dissertation Report* [PDF]. 2011. 116p.
5. Ban Keat Kwa, Dave Hubbard, Monica Smith, *The Value of a PNR Data Warehouse* [PDF]. 2007. 8p.
6. *Amadeus Procuts Blog* [interactive], [viewed on 10 May 2014]. Online Access <http://www.amadeusproductsblog.co.uk>
7. Amadeus IT Group SA, *Qantas cross sell, Case Study* [PDF]. 2012. 8p.
8. *Wikipedia: Usage Share of Operating Systems* [interactive], [viewed on 10 May 2014]. Online Access <http://en.wikipedia.org/wiki/Usage_share_of_operating_systems>
9. *Wikipedia: Android Operating System* [interactive], [viewed on 10 May 2014]. Online Access <http://en.wikipedia.org/wiki/Android_%28operating_system%29>
10. *Wikipedia: iOS* [interactive], [viewed on 10 May 2014]. Online Access <http://en.wikipedia.org/wiki/IOS>
11. *Developer Economics: Pros and Cons – Top 5 Cross Platform Tools.* [interactive], [viewed on 10 May 2014]. Online Access <http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/>
12. *Google Cloud Messaging for Android* [interactive], [viewed on 10 May 2014]. Online Access <http://developer.android.com/google/gcm/index.html>
13. *Local and Push Notification Programming Guide* [interactive], [viewed on 10 May 2014]. Online Access <https://developer.apple.com/library/mac/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Introduction.html>
14. *Phonegap Artwork* [interactive], [viewed on 3 Jun 2014]. Online access <http://phonegap.com/about/artwork/>

Appendixes

1. 15below ltd signed declaration.
2. CD including PDF version of dissertation, .APK (Android) and .IPA (iOS) files.