Ticket Client

Ticket Client is an application for my dad that's managing clients, work tickets and appointments in the IT sector. It is therefore divided into three main parts, where everything is governed by storage in a database and an API specifically for images. Besides storing very complex data, the application helps you with some features such as sending by e-mail, archiving, discharge contact list and a lot more.

To start, the application is composed of a screen holder that change during navigation and, logically, a navigation bar to switch from main menu, to a main part or to the settings screen.

Main screens

Main screen: The main screen serves as a launch pad for the user. It shows the current date, time and calendar of the day as well as the last and oldest modified tickets. In addition, there is a small text field used as a notepad, which is saved in the phone's file system.

Settings screen: This screen allows the user to change the global configuration and behaviour of the application. You can switch to dark or light theme, setup your e-mail or your per-written messages (all of them saved in the phone) as well as possible jobs on tickets and ticket templates. Also, it doesn't use other sub-screen but when you want to create or manage a ticket template, the screen calls another one from the ticket side (*Creation screen* or *Modification screen*) and enable a special mode called "Template". And finally, when managing possible jobs, due to database sharing, you can't delete one if tickets use it.

Main fragments

Sorting fragment: This fragment is used by listing screens to make navigation and researching easier. It implements a search bar (That uses title of elements) and alphabetical or date-based sorting options, both with their inverted version. Also, screens that use it can add boolean options associated with a field of listed elements.

Table field fragment: This one, used in form screens, is comparable to little table where you can add or removes lines and modify their values. There are associated with a database model and matches its functionalities like mandatory, unmodifiable or default values.

Client side

The first part consists of storing client profile that saves main information about such as the name, the phone number, the address and more.

Screens

In this section, there are 2 screens.

Listing screen (Main): The main screen is obviously the listing of all clients with searching options.

Client screen: This one is special case; it has two modes: Creation and Modification. Respectively, it allows you to create a client in a complete way and, after the creation, it enables you to modify data of a client (If fields allow it) or just visualize they.

Fragments

Also, there are 2 fragments used to help implementation of the client side in other screens.

Viewing fragment: This fragment is displayed in the *Listing screen* to see a summary or when creating or modifying a ticket. It shows the label, the phone number and an icon to make a call.

Quick creation fragment: Finally, this one is only used when you create a ticket and you want to associate a new client quickly, without having to switch of screen. It allows you to create a client by setting only the first and last name and the phone number.

Functionalities

The Client side overrides the default Android calling system to display all information the app has (Only if the phone number is known by it) and so permits to discharge the "contacts" of all these people.

In addition, the application provides some pre-written messages to save time while working, especially for sending emails or SMS (Definable in settings).

Database tables

Clients (client)

Name	Type	Default value	Specification

identifier°	String of length 256		Auto unique hexadecimal value
firstName*°	String of length 48		
lastName*°	String of length 48		
label	String of length 64	*firstName* and *lastName*	
phone	String of length 10	Empty string	
email	String of length 320	Empty string	
address	String of length 64	Empty string	
postal	String of length 5	Empty string	
city	String of length 48	Empty string	

Ticket side

The second part involves storing all work tickets, i.e., the tasks issued by the clients, to be performed on their products. So, on these tickets you can find all information is needed to work well and be organized, such as the job list, the deposit list, images, signatures and a lot more.

Screens

The ticket side contains almost the same screens as the client side. Just two are added because of the from complexity.

Listing screen (Main): The main screen is obviously the listing of all tickets that's not a template, with searching options (An option is added to remove finished tickets).

Ticket screen: This one is special case; it has two modes: Creation and Modification and a boolean option: Template. Respectively, it allows you to create a ticket in a complete way, next, after the creation, it enables you to modify data of a ticket (If fields allow it) or just visualize they, and finally it permits to manage his templates,

where certain field are removed if their associated database table don't have the "template" field.

In all screens, you have the fields that the main database indicates but also *Table Field* for the other databases. Besides, only dates, identifiers and "template" fields aren't displayed and the client identifier field have a research bar implemented to make easier the association or the *Quick creation fragment* of the client side.

In addition, the images field permits to you to manually sort them, where the first is considered as the thumbnail, and the app handles the API's limit of images per ticket.

Images screen: This screen is in reality 2 screens, one that takes pictures and the other that allows to see them, from the Android base. They will serve to take an image for the ticket, where the first one is considered as the ticket thumbnail, see them and also show the signature.

Signature screen: Each ticket can have a signature from the client, at the start and the end of the ticket. So, this screen permits to create one, from a drawing that's converted to an image file. Also, if the ticket isn't finished, you can't create an ending signature.

Fragments

In contrary of the client side, this section has only one fragment.

Viewing fragment: This fragment is displayed in the *Listing screen* to see a summary. It shows the title, the product, the thumbnail, an icon that either allows to close it or show that's closed, and the last modification date.

Functionalities

Same as the Client side, there are automatic pre-written messages to send a report of the ticket by e-mail or SMS (In addition of send it to the clients, the application also sends it to the app user).

Besides, when you have finished a ticket, you can close it to make it unmodifiable. If you don't want to clutter up your current ticket list, you can delete it and make a PDF archive of it and save it wherever you want.

Moreover, if you have a repetitive ticket format or you have to be quick, you can create a template (From the settings) and use it in the creation of a ticket to fill up fields automatically with your own default value.

Database tables

Tickets (ticket)

Name	Type	Default value	Specification
identifier°	String of length 256		Auto unique hexadecimal value
client*°	String of length 256		Client identifier
title*	String of length 64	*product*	
product*°	Integer		0 = Laptop 1 = Computer 2 = Tablet 3 = Printer
price	Integer	0	Price of the workmanship and the time
information	String of length 256	Empty string	
external	Boolean	false	Is only with an appointment
bin	Boolean	false	Is destinated to bin
creation°	Timestamp	Current timestamp	
modification	Timestamp	Current timestamp	Change on each modification
completion	Timestamp	NULL	
template°	Boolean	false	

Current jobs (currentJob)

Name	Type	Default value	Specification
ticket°	String of length 256		Ticket identifier
type*°	Integer		
complement°	String of length 256	Empty string	
template°	Boolean	false	

Jobs (job)

Name	Type	Default value	Specification
identifier°	Integer		Auto incrementing unique value
name*	String of length 32		
price°	Integer	0	
complement°	String of length 16	NULL	Complement type NULL = No complement
default°	String of length 256	NULL	Default complement value

Deposits (deposit)

Name	Type	Default value	Specification
ticket*°	String of length 256		Ticket identifier
type*°	Integer		0 = Power

			1 = Battery 2 = Holder 3 = Sensor 4 = Cable
template°	Boolean	false	

Profiles (profile)

Name	Type	Default value	Specification
ticket*°	String of length 256		Ticket identifier
password°	String of length 24	Empty string	
pin°	String of length 8	Empty string	

Purchases (purchases)

Name	Type	Default value	Specification
ticket*°	String of length 256		Ticket identifier
product°	String of length 64	Empty string	
price°	Integer	0	

Calendar side

The last part corresponds to the calendar. It will allow you to see the creation and completion dates of all tickets, in a readable format and also to make appointments with a client and some associated tickets.

Fragments

Calendar of the day: This fragment is only used in the main menu and consists on display the created, finished tickets, and appointments of the day.

The "appointment" side of this section is not yet implemented and will be in future updates.

Images API

This API is only used to upload images from the application to a web server and retrieve them. For authentication, just the sending of a permanent bearer token (Generated by the API) is needed. Besides, the system uses the JSON format, except for image sending. Also, you have to know that the API only accepts image of type *png*, *jpg* or *jpeg*.

Routes

Name	Method	Description
/<ticket>	Get	Retrieves all links to images of **<ticket>**.
/<ticket>/<id>.<extension>	Get	Retrieves an image of a **<ticket>** based on its file name **<id>.<extension>**, where the id is a three digit number with left padding zeros.
/<ticket>	Post	Uploads an image to the **<ticket>** and assigns it either an index that places it at the end of current image list or the specified index at query string field **index**. The body data must be *multipart/form-data* encoded and the body must have an **image** field that contains the image.

		Also, the arbitrary maximum number of images per ticket is 1000.
/<ticket>	Put	Changes the image list order of **<ticket>** by the new one, specified in the body, as a list of the ticket's image complete file names.
/<ticket>/signature	Get	Retrieves all links to signatures of **<ticket>**.
/<ticket>/signature/<type>.<extension>	Get	Retrieves a *start*ing or *end*ing signature of the **<ticket>** based on its file name **<type>.<extension>**.
/<ticket>/signature/<type>	Post	Uploads a *start*ing or *end*ing signature, specified by **<type>**, to the **<ticket>**. The body data must be *multipart/form-data* encoded and the body must have an **image** field that contains the image. Also, if it already exists, the uploading fails.
/<ticket>	Delete	Deletes all images and signatures associated to the **<ticket>** or just the image at specified **index** in query string.