

Critique of TaxiCloudToo

What is the software supposed to do?

TaxiCloud Too is required to be an iPhone application for iOS7 that allows the client to manage the bookings and drivers for a taxi company in a 'dispatcher' role. This includes:

- Adding, modifying and deleting bookings
- Adding, modifying and deleting drivers
- Assigning bookings to drivers

The app should also interface with the TaxiCloud TNG project that was produced concurrently with TaxiCloud Too to provide the database and API.

What does it do?

TaxiCloud Too, based on attempted usage, is unable to perform any of these tasks. The application crashes when trying to enter customer details for a booking, and options still exist for functions that do not exist such as Drivers and Customers.

Requirements and Specification

The requirements for this project have been split into 2 sections – the Design Requirements, those that have been decided upon or come to by the developer, and Project Requirements, which have been given to them by the client. No obvious mention of specifications, although it could be argued that this is somewhat covered by 'Design Requirements'.

Premise of the project was to provide a mobile application to access the TaxiCloud web interface system, being refactored and developed concurrently with this project. There were a number of issues regarding the previous iterations of the TaxiCloud system and the aim was to be able to address these with the TaxiCloud TNG team and create a homogenous system. However, it appears that there was a breakdown in communication between the two as there are still issues regarding differences in authentication and protocols.

Design and Implementation

The design of the application is poor, from planning right through to implementation. There is a severe lack of relevant and insightful documentation which makes it very difficult to provide a holistic critique of their methods – which is in it's own right a critique of their methods. They provide a number of documents describing in limited detail some of the technologies and concepts they aimed to implement in the project. These are 'User Levels', 'Networking', 'Google Authentication', 'Test Server' and 'TapKu'.

'User Levels' details the differences between the proposed levels of user access – the original plan called for only a Dispatcher view but views for Drivers and Customers were subsequently required and removed from iterations.

'Networking' and 'Google Authentication' provide overviews of the networking and authentication techniques used by the developers in their application.

'Test Server' explains about the creation of a barebones test server for use with the application due to issues with both TaxiCloud and TaxiCloud TNG that were not resolved by the development team.

'TapKu' describes in little detail about the TapKu library and what information needs to be passed to it for it to work correctly.

UI Design is covered in a very short document that details the use of storyboards to influence design, rather than discussing the design or the design process in any detail.

Testing

Although documents for a 'Test Plan' and 'Test Data' have been provided, there are no reliable or recognizable testing criteria shown to be met or even produced. Any testing seems to have been conducted manually by the developers and is limited to what they can see on the screen. This is shown when trying to use the application itself – even with a very cursory first usage a number of faults and exceptions were encountered. Options still exist for the non-existent Driver and Customer views, and when trying to add a new booking XCode was forced to go through a debugging process and the program failed. A number of the manual tests that were carried out are marked as having 'failed'.

Evolution

The severe lack of documentation and comments in the code make the evolution of this code exceptionally difficult. There exist no examples of UML documentation, severely compromising further development due to a lack of information for a new development team to work from. It will require a complete overhaul of the current documentation structure and the production of UML Class, Sequence and State diagrams for the current iteration before progress can be made.

User Interface

Overall design is aesthetically pleasing, the colours compliment the TNG Web Interface and it is generally uncluttered. Specific issues are noted on a page by page basis.

'Bookings' Page: Design is nice, although there are no bookings for us to be able to accurately critique the design of the page when there are bookings. Basic empty screen is acceptable.

'Calendar' Page: Date in title is too close to the Notifications Bar, overlaps and does not look pleasing. Use of a 'day list' view with no option to change. Unable to scroll through calendar (fixed view with current time as anchor).

'More' Page: Superfluous page considering that the Customer, Drivers and Companies pages do not exist. Logout need not be a 'setting' as such and could be handled in the same way that 'Edit' and '+' are on the Bookings page. No opportunity to change any settings whatsoever.