Advanced Programming for Mobile Devices - Proposal

James Donohue - james.donohue@bbc.co.uk

Aims

Overview

This proposal is for an iOS mobile application ('app') which enables owners of motorcycles and scooters to 'crowdsource' the location of their vehicle in the event of it going missing using a combination of Bluetooth Low Energy (BLE) beacons and the Global Positioning System (GPS).

Context

Thefts of motorcycles and scooters are growing sharply. Nearly 15,000 bikes were stolen in the year 2016-7 in London alone (BBC, 2017), an increase of 30% on the previous year. As a recovery measure, owners are advised by authorities to apply ultraviolet markings (City of London Police, 2016). Another option is a GPS tracker which can be used to report the vehicle's location to a central service and therefore recover it if stolen, however prices for these devices start at £300 (BikeTrac, 2017) and usually also entail a monthly subscription charge, making them not affordable for many riders.

iBeacon is a Bluetooth Low Energy (BLE) technology developed by Apple that enables iOS apps to recognise when they have entered or left the physical region around a compatible device that is generating iBeacon advertisements (Apple, 2014). In contrast to GPS trackers, iBeacons are a low-cost solution and require no subscription.

Proposal - 'FindMyBike'

When physically attached to a motorcycles/scooter and associated with details of the owner and the make and model of the bike, an iBeacon may be used with the proposed app 'FindMyBike' to determine if the bike is nearby.

If a sufficiently large number of iOS app users are present in a densely populated area such as central London, the chances of detecting an iBeacon attached to a missing vehicle becomes high enough to make it a cost-effective alternative to GPS trackers by 'crowdsourcing' the absolute location of the missing vehicle.

Figure 1 describes the desired behaviour of the proposed application. The sequence of app usage can be summarised as:

1. The user installs the FindMyBike app on their iPhone

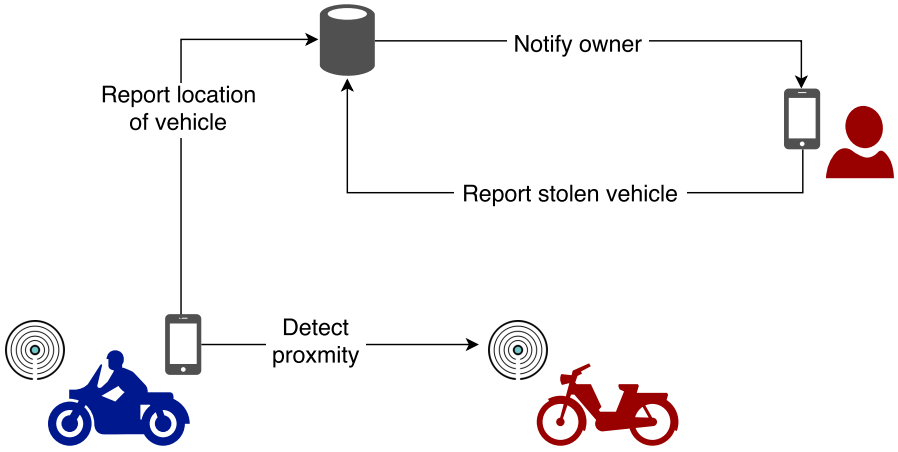

Figure 1: Overview of solution

2. The user attaches a compatible iBeacon device to their bike and registers its identifier with the app
3. If the user's bike is stolen, they use the app to report it to the FindMyBike servers
4. As other FindMyBike users moves around in their normal life, if they enter a physical region near the stolen bike, a notification is sent to their phone
5. On receiving such a notification, another user opens the app to see the make and model of a stolen bike they are near
6. This user optionally shares their absolute location (determined via their phone's GPS receiver) with the owner of the stolen bike

Target user group

The target market for this app are motorcycle and scooter owners living in densely populated urban areas with a known motorcycle crime problem such as central London. Users will be required to own devices running iOS 10 or later (representing 86% of the iOS device market (Apple, 2017) in order to balance availability of the latest APIs with the maximum potential installed user base.

Platform and technologies

The initial target platform for development is iPhone 6 and later. The report will also explore the possibility of adapting the app for Apple Watch.

All iOS code for the project will be written using Swift 3.1, to give an opportunity to explore and reflect upon recent trends in the Apple ecosystem.

The app will explore iOS architectural features including but not limited to

- Core Location framework
- User Notifications framework
- UIKit framework
- Foundation framework

Server component

A key part of the project is a network-accessible server that is responsible for:

- storing details of registered users and bikes
- notifying app users about bikes that are reported missing
- collecting location data reported by app users about detected missing bikes
- notifying the owner of a missing bike that it has been detected

Fully creating such a server is out of scope for this project, however a simplistic implementation will be provided in order to validate the design and behaviour of the iOS app.

Measurement

The success of the project will be measured in a number of ways:

- compatibility with target platform and devices
- demonstration that the app would meet Apple's criteria for inclusion on the App Store

Following inclusion on the App Store, the following criteria should be considered when considering how effective the app is:

- number of download on the App Store
- number of owners who have registered their bikes the app
- number of missing bikes reported as later recovered using the app.

Blueprint

Figure 2 shows a partial design blueprint in the form of UI mockups of two key scenarios in the usage of the app.

In the top scenario, a normal user is able to update their bike details and also view information about iBeacons in range that are attached to bikes which have been reported missing by other users. In the bottom scenario, the owner of a missing bike receives a 'push' (remote) notification from the server that their bike has been located. Upon opening the app they see the precise location submitted by the first user.

Design and development pattern

Model-View-Controller

The envisaged architectural pattern for the app is Model-View-Controller (MVC). MVC was formulated by Trygve Reenskaug in 1978 and later developed by others at Xeroc PARC (Reenskaug, 2012), and has since been applied to a wide range of application platforms. Freeman (2015) states that one aim of MVC is to 'simplify development, testing and maintenance'. Indeed, Apple's developer documentation (Apple, 2012) indicates that core iOS platform technologies often assume custom app objects will conform to the MVC pattern.

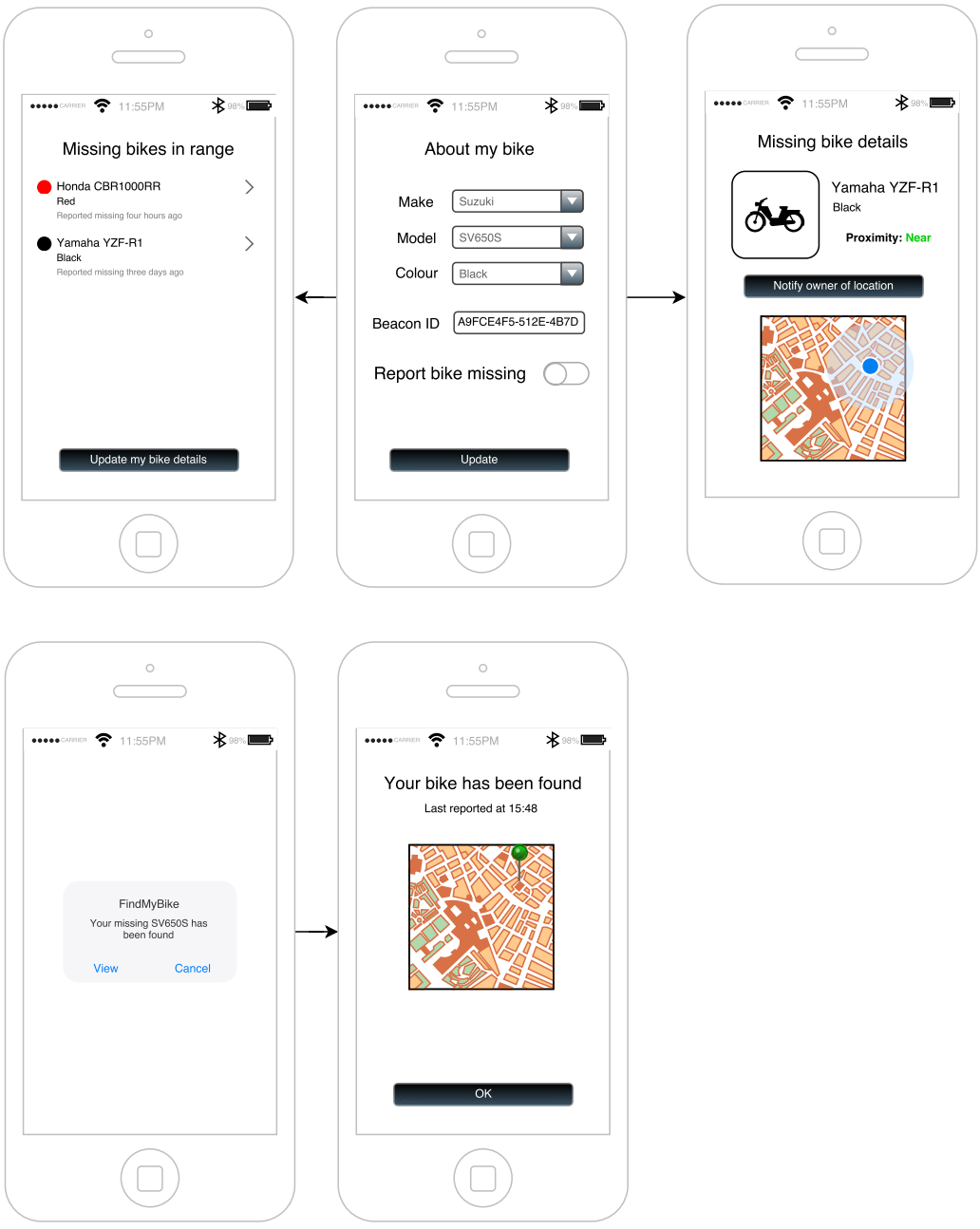

Figure 2: Design blueprints showing scanning mode (top) and location of missing bike (bottom)

MVC can be seen as a layered model in which different applications concerns are separated. The model is an abstract representation of some data relevant to the application (for example, the registration details of a vehicle) along with operations for manipulating it, while the view corresponds to one or more user interface components that display data to the user. This implies that multiple views may be attached to the same model. The controller acts as intermediary between these layers and 'defines the way the user interface reacts to user input' (Gamma et al., 1994). By decoupling data and presentation, the resulting app code can be more reusable and extensible (Apple, 2012).

One possible variation on the pattern is to also define a 'view model', which may be defined as a restricted or modified set of model data that is used by the view (Freeman, 2015), but this additional separation is unlikely to be necessary for the relatively simple data used by the proposed app.

Development approach

As some parts of the solution are not yet fully defined, an iterative development method is proposed. Iterative development implies a process of evolutionary advancement as opposed to a sequential, document-based one (Larman and Basili, 2003). It also entails a *production prototype* that 'evolves as the project team learns more about the solution' (Wysocki, 2014).

The selection of this method is based on the following assumptions:

- There is a single designer/developer working on the project
- There will be no change in team members during the project
- The designer/developer also acts as the primary 'client' of the project (with the University as a secondary client)
- Changes in scope/approach are possible but will be be determined solely by the designer/developer

Iterative project management models are well suited to learning and discovery, where a solution is known but not to the required depth, and some scope changes are expected (Wysocki, 2014). Although for some projects the need for a very involved client is a challenge, this problem does not apply here.

Development timeline

Figure 3 is a simplified Gantt chart shows the design and development plan for the project. Although Gantt charts are normally associated with the 'waterfall' software development model, it is used here as a way of illustrating the phases of development, with each phase in the left column representing an incremental iteration.

Task owners are not specified as only a single designer/developer is involved.

Risk register

A number of possible risks for this project have been identified, which are given below in the form of a simplified risk register in the style of PRINCE2. The objective of such a register is "to identify, assess and control uncertainty and, as a result, improve the ability of the project to succeed" (Barker, 2013).

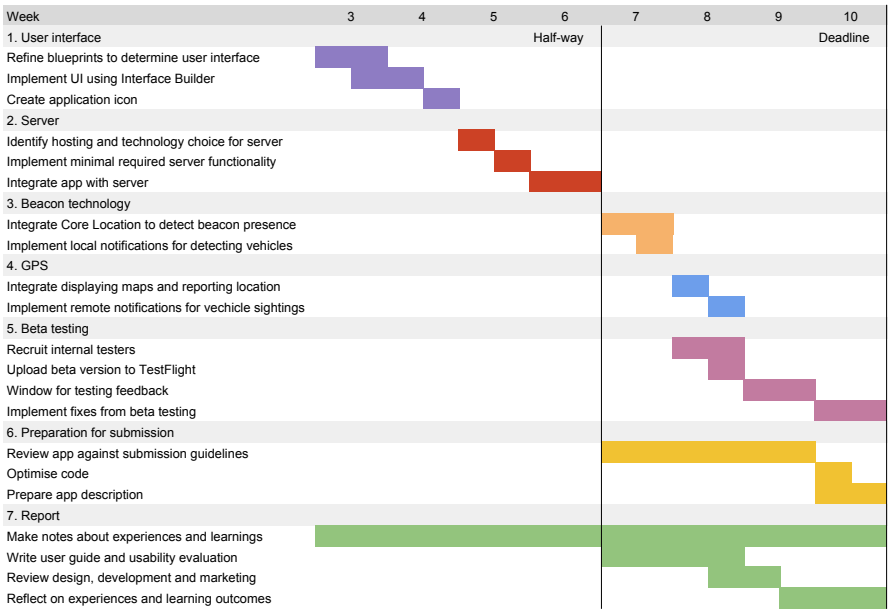

Figure 3: Development timeline Gantt chart

ID	Description	Probability	Impact	Level of Risk
1	Limited signal range of iBeacon devices makes proximity detection impractical.	Possible	High	High
2	Insufficient users install the app after release to provide necessary level of coverage	Possible	High	High
3	Lack of Bluetooth support in iOS Simulator slows down development by requiring use of real physical devices for testing	Probable	Low	Medium
4	The app fails to comply with submission guidelines for the App store	Unlikely	High	Medium

Based on the risks above, the following mitigation strategies are suggested:

- Early study of app store submission guidelines and ongoing review [4]
- Testing with a range of different compatible mobile devices and iBeacons [1, 3]
- Development of a thorough marketing strategy to maximise downloads [2]

References

Apple (2017) *App Store*, Available at: https://developer.apple.com/support/app-store/ (Accessed: 29 July 2017).

Apple (2012) *Concepts in Objective-C Programming: Model-View-Controller*, Available at: https://developer.apple.com/library/content/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html (Accessed: 29 July 2017).

Apple (2014) *Getting Started with iBeacon*, Available at: https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf (Accessed: 29 July 2017).

Barker, S. (2013) *Brilliant PRINCE2*, Harlow, England, Pearson International.

BBC (2017) *The moped and scooter crime wave that has swept London*, Available at: http://www.bbc.co.uk/news/uk-40731485 (Accessed: 29 July 2017).

BikeTrac (2017) *How much does BikeTrac cost?*, Available at: https://biketrac.co.uk/#pricing (Accessed: 29 July 2017).

City of London Police (2016) *Motorcycle Theft is on the Increase*, Available at: https://www.cityoflondon.police.uk/news-and-appeals/Documents/Motorcycle%20Theft%20leaflet.pdf (Accessed: 29 July 2017).

Freeman, A. (2015) *Pro Design Patterns in Swift*, New York, NY, Apress.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994) *Design Patterns: Elements of Reusable Object-Oriented Software*, Upper Saddle River, NJ, Addison-Wesley.

Larman, C. and Basili, V. R. (2003) *Iterative and Incremental Development: A Brief History*, Available at: https://www.cs.umd.edu/~basili/publications/journals/J90.pdf (Accessed: 29 July 2017).

Reenskaug, T. (2012) *MVC - Xerox PARC 1978-79*, Available at: http://heim.ifi.uio.no/~trygver/themes/

mvc/mvc-index.html (Accessed: 29 July 2017).

Wysocki, R. K. (2014) *Effective Project Management: Traditional, Agile, Extreme*, Seventh edition, Indianaopolis, IN, John Wiley & Sons.