Final Year Project 2015/16
Smartphone Based Vehicle Recorder

Appy Driving

Interim Report

Student: Chung Man Ip, Clement
Supervisor: Dr H.Y. Chung, Ronald

Date: Jan 24, 2016

Project Background

Introduction

Vehicle Recorders, also known as Dash Cams, are becoming more and more popular in the Auto Accessories Industry. Currently, there is already a wide variety of devices existing on the market, however, most, if not all, of them has really similar characteristics such as having a camera which supports looped recording, with an external storage card as its primary storage. Although the installation cost for a vehicle recorder is not cheap, there are great reasons for investing in this gadget. For instance, the video footage captured can be used as video evidence in cases of claiming accident insurance, or even proof of innocence in legal inquiries. This provides a solid proof other than words from the driver, victim, or other witnesses, and is also a good explanation for the fact that vehicle recorders are ubiquitous in Russia, which has recorded the highest number of car accidents and fatalities in 2009[1].

With the unstoppable advancement in the Smartphone Industry, both the hardware and software capabilities of them improve rapidly in recent years. Some developers have started to implement camera-related apps to bring the vehicle recorder features into smartphones. They made use of the built-in camera and internal storage of the phone to virtualize a vehicle recorder in the driver's smartphone, making the installation of vehicle cameras much easier and cheaper. However, a simple camera application does not fully utilize the true power of smartphones, especially in regards with the aspect of social networks.

In this information era, smartphones are actually a portable hotspot for data sharing through the Internet. Push notifications and social networks have established various channels to send and receive information via the Internet. On the other hand, hardware capabilities offer a better user experience as well as value-added functions compared with a conventional vehicle recorder. For instance, with the data gathered by an accelerometer and Global Positioning System (GPS), mobile applications could provide extra useful information such as car speed and current position.

[1] http://www.forbes.com/2009/05/19/dangerous-countries-roads-lifestyle-travel-dangerous-

The aim for this project is to design and develop a Smartphone Vehicle Recorder, with add-on features that are designed for general drivers.

Conventional vehicle recorder

Vehicle recorders were introduced to police cars in Texas in the 1980s, at that time, the primary function was to capture everything ahead of the vehicle. With the fading out of VHS cassettes, the size of the vehicle recorder has become much smaller so that installation became much easier as well. In the 2000's, vehicle recorders entered the domestic auto market, getting more and more common among general drivers. Drivers who look for extra protection would invest in this gadget in order to guarantee that there is a video footage of what happened when there are unfortunate cases of traffic accident.

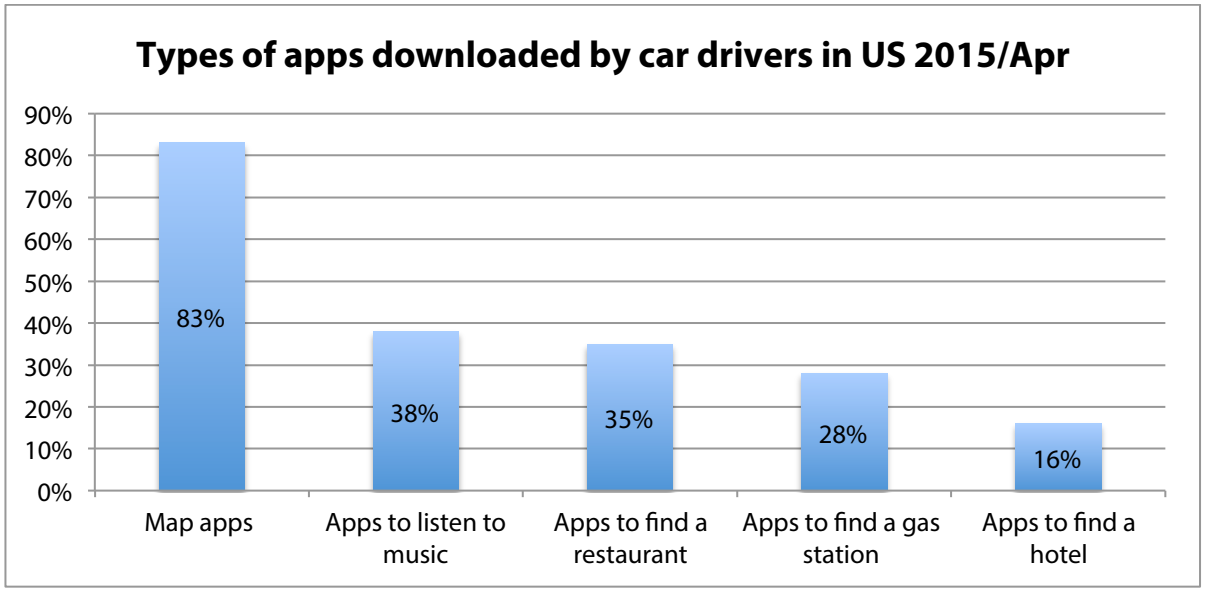
Variations among different manufacturers of entry-level vehicle recorders are just mainly between camera quality and size of the recorder[2]; with some suggesting a better night view camera but with relatively larger body, while the others offer a stealth-looking recorder with a smaller size so that it will not block the view of the drivers. As regular recorders, those with a camera capturing movements ahead and storing footages in their internal storages, has become saturated in the market, manufacturers start to implement more value-adding functions to their products. In an effort to differentiate themselves, products with multiple cameras capturing front and back views, and products equipped with GPS or G-Sensor to support geotagging and motion detection start to appear on the market.

It is troublesome to configure the camera settings by just observing the LED light indicator, or to fine-tune the camera to the desired angle without a preview display, hence, manufacturers have started to product products which come with a small LCD panel. This design allows the drivers to check whether the camera is really working during road trips, thus having a peace of mind that every moment is being captured.

[2] http://dashboardcamerareviews.com/dash-cam-comparison-table/dash-cam-comparison-affordable/

Current Market Analysis

Of over one and a half million applications available on the App Store, applications related to vehicle recording can be categorized into two main streams- one being an individual application that offers a Dash Cam function, while the other is developed to work cooperatively with a physical vehicle camera. However, only a few of the applications could leverage the true power of smartphones, such as network accessibility and multi-tasking capability, to bring added values for the users.



According to a recent research shown above[3], most drivers in the United States are looking for smartphone applications that offer a map function, while less popular functions are music-playing and place-finding. This research indicates that drivers do want to utilize their smartphones to assist themselves with their driving, hence providing additional information to them.

To conclude, there is clearly a gap between existing products currently on the market and user expectations. Undoubtedly, standalone applications are sufficient to fulfill the need of a particular function, but the needs may vary based on different locations and time factors. In order to bridge the gap between the existing market and user expectations, it would be advantageous to have an application that can satisfy various drivers' needs regardless of where and when they are, and utilize the full capabilities of smartphones to enhance the user experience.

[3] http://www.statista.com/statistics/428070/types-of-apps-downloaded-by-car-drivers-us/

Project Objective

"Appy Driving" – the name of the application of this project, aims to offer an all-in-one platform which includes multiple functions to satisfy the general drivers' needs. Various functions will be delivered as modules that run on top of the core framework. Likewise, there will be multiple applications running on the smartphone platform; hence the name "Appy" literally means functions with modular design like an application.

Features

1. User Customization

Users can organize their own application layout according to their preferences. Each module can be independently shown or hidden, and display in whatever order within the core framework. There are two possible layouts for users, dashboard view or single-module view, to decide the content to be shown. The former view provides an overview of all running modules, while the latter focuses on one particular module. If some of the modules will not be used for a long period of time, the users can simply swipe on the particular module to hide it in order to have more free space for other working modules.

2. Optimal Video Quality

For an accident video footage captured in just a couple of seconds, drivers would want to capture every single pixel as clear as possible, in order to retain a strong proof as an evidence. However, the higher video quality the more space consumed, and the other applications' service would be affected if the video files use up all the storage space. Appy will automatically adjust the best video quality according to the storage space available and hardware capabilities of the users' phone.

3. Social Connectivity

Driving is not alone anymore with social features implemented. Users can share their video footages with friends or even other drivers, discuss on the experiences encountered on roads and highways. This sharing platform gathers comments from drivers, thus ultimately contributes to the resources base of this application. For instance, users can suggest alternative RSS news feed to others, if the other drivers find it useful, they can up vote the content as well as the contributor.

Core Functions

The table below briefly describes each module and its functions:

Modules	Functions
Vehicle Camera	• Looped Recording • Read-only Flags
Navigation	• Way-finding • Travel Distance Log
Traffic News	• Instant Traffic News • Support Custom RSS Feed
Fuel	• Nearby Fuel Station • Cost Estimation
Social	• Share Videos/Routes • Updated News

Project Methodology

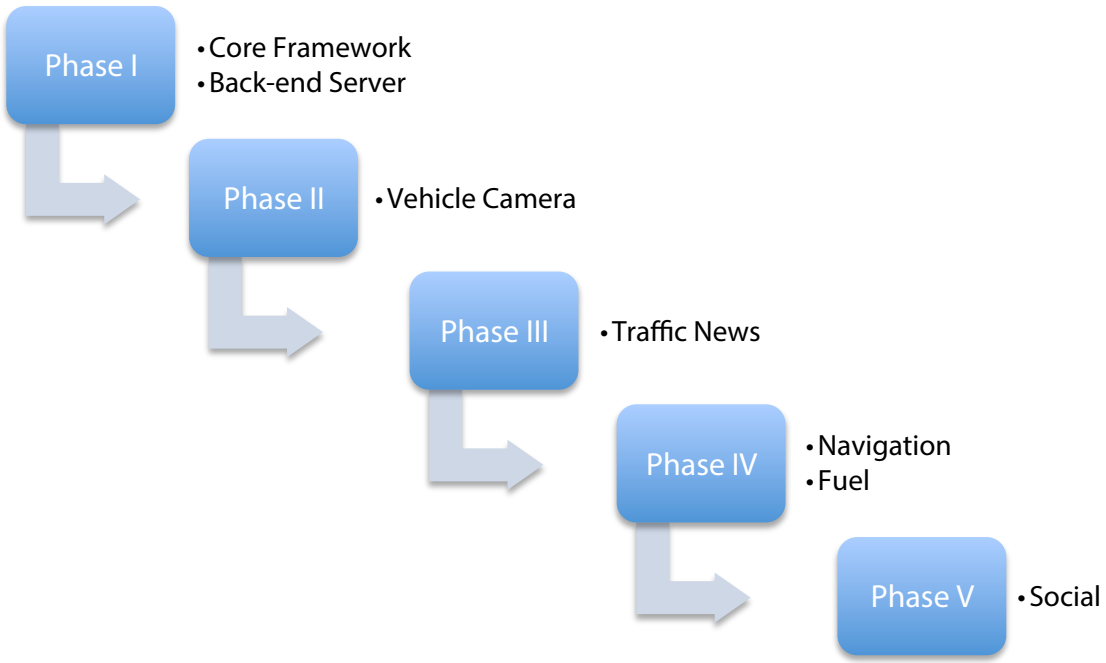
Implementation

The whole project consists of a front-end mobile application as well as a back-end server. For the front-end mobile application, it would include basic vehicle camera functions and value-added functions with modular design in order to enhance the overall user experience. In the smartphone industry, iOS and Android are the two most common operating systems, with the number of applications available in the App Store on par with the Play Store. This project will be built on the iOS platform targeting iOS 8.0 first, due to lesser variations regarding both hardware and software aspects, so that the quality and user experience controls would be easier to manage. The development of the application for Android platform would be discussed in the future development.

For the back-end server, it would be responsible for the content management service of the application, and also as a database for the primary storage of information. Regarding the development platform, Ruby on Rails would be a desirable candidate in terms of its scalability and flexibility, as it offers advantages both in the database management and migration. There will be two environments, development and production respectively, for the deployment of the back-end server. Both environments will be almost the same with regards to the design, structure and functions. However, the development environment will be mainly for internal testing only, while the production environment will be solely for public usage. This practice can stabilize the deployment since the external database is being isolated with the internal one, hence the public users' data will be safe even if there is a critical failure encountered in the testing stages.

Project Management

Since the application is based on a modular design, the whole project will adopt the Phased Methodology in order to design and implement the system. The development cycle will be categorized into five phases, with the earlier phases focusing on the core framework and basic functions such as working as a vehicle recorder while the latter phases would be value-adding add-ons built on top of the core framework. Each module can be tested independently and thus guarantee the quality of the deliverables.



Project Schedule and Milestone

Tasks	Pre-decessor	Expected Start Date	Expected End Date	Duration (days)	Milestone
A: Meeting with supervisor Preliminary requirements gathering	-	Sep 14 2015	Sep 24 2015	10	
B: Finalize scope of project	A	Sep 24 2015	Oct 4 2015	11	Project Plan Project Webpage
C: Phase I Coding and Testing	A	Oct 4 2015	Oct 31 2015	28	
D: Phase II Coding and Testing	C	Nov 1 2015	Nov 30 2015	30	
E: Examination Break	-	Dec 1 2015	Dec 23 2015	23	
F: Phase III Coding and Testing	D	Dec 23 2015	Jan 18 2016	27	
G: Interim Testing and Evaluation	D, F	Jan 18 2016	Jan 24 2016	7	Interim Report Product Prototype
H: Phase IV Coding and Testing	F	Jan 18 2016	Feb 25 2016	39	
I: Phase V Coding and Testing	H	Feb 25 2016	Mar 31 2016	36	
J: Final Testing and Evaluation	H, I	Apr 1 2016	Apr 17 2016	17	Final Report Finalized Product
K: Project Exhibition and Presentation Preparation Documentation	J	Apr 18 2016	May 3 2016	16	Project Exhibition Materials

Implementation Detail

Phase I

In this phase, the objective is to implement a back-end server as well as design and create the core framework of application itself. For the former part, a server that running Ruby on Rails was set up via the department of Computer Science, HKU. The system and database schema was initialized and ready for phase III development in the near future. On the other hand, UI design was done and implemented as a core framework of the application. Instead of using traditional Navigation Controller model with several segues pointing to other Model-View-Controller (MVC) instances, Appy adopts separate MVC structure in order to enhance the user experience. With isolated MVC instances, users are able to navigate different functions by simple swipes. Moreover, it is more maintainable in terms of testing and maintenance as a result of separate MVC with minimal amount of dependencies. Each MVC instance contains a *.xib* file as view, *.swift* file as controller and other data formats as model.

Phase II

Stepping into second phase of deployment, the main function in this phase is to implement the camera recording function. In order to make use of built-in camera devices of Apple devices, there are two classes available in Swift 2.0 to enable the vision: UIImagePickerController and AVCaptureSession. UIImagePickerController manages user interfaces for taking images and movies, and for choosing the media from user library (i.e. Camera Roll in Apple devices). Comparatively, AVCaptureSession is just an interface to coordinate the flow of data from AV input sources to outputs, without the media picking function and built-in camera function like HDR and flash control. General users may be more familiar with former interface as it is the same as iOS built-in camera app, but Appy camera is implemented according to the AVCaptureSession as it suggests a much more powerful camera interface than UIImagePickerController.
In AVCaptureSession, input devices available to the class are front and rear camera, microphones, etc. In order to perform a real-time image taking or video shooting, after creating an instance of AVCaptureSession the application requires handling various input

devices and adding them into the current session. After that, configure appropriate output format and preset to manage the quality as well as output format.
The quality of video will be adjusted automatically according to the preferred space set by the user, to achieve this several video preset are used as follows:

sessionPreset Properties

AVCaptureSessionPresetHigh	Suitable for high definition video output
AVCaptureSessionPresetMedium	Suitable for sharing over WiFi
AVCaptureSessionPresetLow	Suitable for sharing over legacy cellular network
AVCaptureSessionPresetPhoto	Suitable for high resolution photo output

Future Development

In the future, the traffic news function would be implemented after phase II development. Within this section, the back-end server implemented before will become a database and data retriever to obtain real-time traffic news available on the Internet. RSS Feed technology would be used to obtain a *.xml* file including news and other related information. On the client-side, latest traffic news would be pushed to Appy and appear like a regular notification.