**Department of Information Science and Engineering**

**Tour of Automobile specs using**

**Augmented Reality**

Project members:

1. Raghavendra C 1GA09IS030
2. Rohit R Deshpande 1GA09IS036
3. Karthik B.N 1GA10IS401
4. Nagesh H.R 1GA10IS403

Under the guidance of

**Mrs.Shruthi K.R**

Lecturer,

Dept of ISE, GAT

**Abstract**

**Abstract**

This project aims at developing an Android Application that would have the capability to show information about a car, its technical specification, interiors, exteriors, looks and feel.

Augmented Reality is a live, direct or indirect, view of a physical, real world environment whose elements are augmented by computer-generated sensory input such as sound, video graphics or GPS data. As a result, the technology functions by enhancing one’s current perception of reality. This mobile Application is built on by taking pictures and videos of the particular automobile and creating a sensible presentation (by stitching all pictures). .

While user focuses camera on particular image the Video starts playing the attractions. This Application can also be shared with friends via Face book or via SMS. Also can pull the latest offers from attractions to Users via Face book.

* This Application helps to see specification, points of interests and other exiting Augmented Reality contents.
* It helps to find events, tweets, Wikipedia articles, ATMs, restaurants, user reviews and much more around you.

**INTRODUCTION**

**Chapter 1**

**INTRODUCTION**

* 1. Introduction to Tour of Automobile Specs Augmented Reality

Aims at developing an Android application that would have the capability to show information about a car, its technical specification, interiors, exteriors, looks, feel, colours and ex-showroom price details.

There are no existing application that integrate AUGMENTED REALITY with automobile specs tour.Integrating augmented reality with Android use real life objects, in this case the automobile, images of the auto mobile, the logo and logo of the model of the car as identifiers triggering the respective tours.

1.2 Problem Definition

There were no products or application that catered to this niche market which could provide the complete details of specification of an automobile.

This was the major problem customers use to face they need to go to the respective website and check for the details so we took an initiative in developing a new app which could serve for the purpose for the customers who can easily scan the logo of the car and obtain the necessary details.

* 1. Proposed System

Integrating augmented reality with Android use real life objects, in this case the automobile, images of the auto mobile, the logo and logo of the model of the car as identifiers triggering the respective tours.

This system could be an added advantage as we are integrating the modules and making use of NDK native development kit this also enhances the easiness present in the system as we could reuse the modules that are available present in the c/c++.

Augmented reality - Augmented reality (AR) is a term for a live direct or indirect view of a real world environment whose elements are supplemented with, or augmented by computer generated imagery. The augmentation is conventionally in real time and in meaningful context with environmental elements. Augmented reality and virtual worlds are long standing topics in their own right. MobEA 2010, now in its eighth year will focus on the topic of the convergence between augmented reality, virtual interactivity, and the Web. How can these new interaction paradigms be integrated into the platform of the Web and more importantly evolve to be as open and inclusive as the Web?

The W3Cs Mobile Web Initiative is advancing pervasive mobile computing through the adoption of standards and best practices. Currently, there are diverging technologies for AR, no standards insight, and this could lead to customer confusion and industry gridlock. The objective of this workshop is to provide a single forum for researchers and technologists to discuss the state of the art for Mobile AR.

## Challenges

In this context, challenges for AR and VR research will include, among others:

* How does Web content need to evolve with AR (e.g. extensions to current Web technologies like CSS to support AR)?
* How far can we stretch advanced UI techniques (e.g. 3D navigation) for building more immersive applications?
* What are some of the strategies for monetizing AR and VR?
* What are some of the accessibility issues of AR interfaces?

**Web and AR**

* How does Web content need to evolve for AR (extensions to current Web technologies like CSS) to support AR ?
* How do Web browsers need to evolve to become AR platforms?
* Can the 3D Web be an effective platform for AR?
* Infrastructure for AR

**User interfaces**

* 3D interaction and gestural interface and how these interfaces map onto the Metaphor of the Web
* Advanced display and immersive projection technology
* Haptics, audio, and other non visual interfaces

**Mobile AR**

* Mobile AR development environments and toolkits
* Mobile Tracking & Sensing for AR
* Mobile Presence and cognition in AR
* Mobile Widgets and Mashups for AR
* Media adaptation and content filtering

**Evaluation of AR**

* User studies and evaluation
* Standard Datasets and Benchmarks for AR
* Accessibility of AR interfaces (What does an AR Web site look like for the visually disabled and what extensions would be required to current web accessibility techniques?)

**FUNCTIONAL COMPONENTS:**

* Store image –Initially the pictures of the automobile will be stored in the server.
* Scan Image-Scan the billboard using the camera.
* Display Information-Once the scanned image matches the picture in the server augmented browser will display the information regarding the automobile under scrutiny.

**literature survey**

# chapter 2

# literature survey

# Augmented Reality

Augmented reality (AR) is a live, direct or indirect, view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented ) by a computer. As a result, the technology functions by enhancing one’s current perception of reality. By contrast, virtual reality replaces the real world with a simulated one. Augmentation is conventionally in real-time and in semantic context with environmental elements, such as sports scores on TV during a match. With the help of advanced AR technology (e.g. adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and digitally manipulable. Artificial information about the environment and its objects can be overlaid on the real world.

**Augmented reality** (AR) is a live, direct or indirect, view of a physical, real-world environment whose elements are *augmented* by [computer-generated](http://en.wikipedia.org/wiki/Computer-generated) sensory input such as sound, video, graphics or [GPS](http://en.wikipedia.org/wiki/GPS) data. It is related to a more general concept called [mediated reality](http://en.wikipedia.org/wiki/Mediated_reality), in which a view of reality is modified (possibly even diminished rather than augmented) by a computer. As a result, the technology functions by enhancing one’s current perception of reality.[[1]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-1) By contrast, [virtual reality](http://en.wikipedia.org/wiki/Virtual_reality) replaces the real world with a simulated one.[[2]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-2)[[3]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-3) Augmentation is conventionally in [real-time](http://en.wikipedia.org/wiki/Real-time_computing) and in semantic context with environmental elements, such as sports scores on TV during a match. With the help of advanced AR technology (e.g. adding [computer vision](http://en.wikipedia.org/wiki/Computer_vision) and [object recognition](http://en.wikipedia.org/wiki/Object_recognition)) the information about the surrounding real world of the user becomes [interactive](http://en.wikipedia.org/wiki/Interactive) and digitally manipulable. Artificial information about the environment and its objects can be overlaid on the real world

**Hardware**

Hardware components for augmented reality are: processor, display, sensors and input devices. Modern [mobile computing](http://en.wikipedia.org/wiki/Mobile_computing) devices like [smartphones](http://en.wikipedia.org/wiki/Smartphone) and [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer) contain these elements which often include a camera and [MEMS](http://en.wikipedia.org/wiki/MEMS) sensors such as [accelerometer](http://en.wikipedia.org/wiki/Accelerometer), [GPS](http://en.wikipedia.org/wiki/GPS), and [solid state compass](http://en.wikipedia.org/wiki/Compass#Solid_state_compasses), making them suitable AR platforms.[[8]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-8)

**Display**

Various technologies are used in Augmented Reality rendering including optical projection systems, monitors, hand held devices, and display systems worn on one's person.

#### Computer

The computer analyzes the sensed visual and other data to synthesize and position augmentations.

### Software and algorithms

A key measure of AR systems is how realistically they integrate augmentations with the real world. The software must derive real world coordinates, independent from the camera, from camera images. That process is called [image registration](http://en.wikipedia.org/wiki/Image_registration) which uses different methods of [computer vision](http://en.wikipedia.org/wiki/Computer_vision), mostly related to [video tracking](http://en.wikipedia.org/wiki/Video_tracking).[[41]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-recentadvances-41)[[42]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-42) Many computer vision methods of augmented reality are inherited from [visual odometry](http://en.wikipedia.org/wiki/Visual_odometry). Usually those methods consist of two parts.

First detect [interest points](http://en.wikipedia.org/wiki/Interest_point_detection), or [fiduciary markers](http://en.wikipedia.org/wiki/Fiduciary_marker), or [optical flow](http://en.wikipedia.org/wiki/Optical_flow) in the camera images. First stage can use [feature detection](http://en.wikipedia.org/wiki/Feature_detection_%28computer_vision%29) methods like [corner detection](http://en.wikipedia.org/wiki/Corner_detection), [blob detection](http://en.wikipedia.org/wiki/Blob_detection), [edge detection](http://en.wikipedia.org/wiki/Edge_detection) or [thresholding](http://en.wikipedia.org/wiki/Thresholding_%28image_processing%29) and/or other [image processing](http://en.wikipedia.org/wiki/Image_processing) methods.[[43]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-43)[[44]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-44) The second stage restores a real world coordinate system from the data obtained in the first stage. Some methods assume objects with known geometry (or fiduciary markers) present in the scene. In some of those cases the scene 3D structure should be precalculated beforehand. If part of the scene is unknown [simultaneous localization and mapping](http://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping) (SLAM) can map relative positions. If no information about scene geometry is available, [structure from motion](http://en.wikipedia.org/wiki/Structure_from_motion) methods like [bundle adjustment](http://en.wikipedia.org/wiki/Bundle_adjustment) are used. Mathematical methods used in the second stage include [projective](http://en.wikipedia.org/wiki/Projective_geometry) ([epipolar](http://en.wikipedia.org/wiki/Epipolar_geometry" \o "Epipolar geometry)) geometry, [geometric algebra](http://en.wikipedia.org/wiki/Geometric_algebra), [rotation representation](http://en.wikipedia.org/wiki/Rotation_formalisms_in_three_dimensions) with [exponential map](http://en.wikipedia.org/wiki/Rotation_matrix#Exponential_map), [kalman](http://en.wikipedia.org/wiki/Kalman_filter) and [particle](http://en.wikipedia.org/wiki/Particle_filter) filters, [nonlinear optimization](http://en.wikipedia.org/wiki/Nonlinear_optimization), [robust statistics](http://en.wikipedia.org/wiki/Robust_statistics).

## Applications

Augmented reality has many applications, and many areas can benefit from the usage of AR technology. AR was initially used for military, industrial, and medical applications, but was soon applied to commercial and entertainment areas as well.[[45]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-45)

### Archaeology

AR can be used to aid archaeological research, by augmenting archaeological features onto the modern landscape, enabling archaeologists to formulate conclusions about site placement and configuration.[[46]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-46)

Another application given to AR in this field is the possibility for users to rebuild ruins, buildings, or even landscapes as they formerly existed.[[47]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-47)

### Architecture

AR can aid in visualizing building projects. Computer-generated images of a structure can be superimposed into a real life local view of a property before the physical building is constructed there. AR can also be employed within an architect's work space, rendering into their view animated 3D visualizations of their 2D drawings. Architecture sight-seeing can be enhanced with AR applications allowing users viewing a building's exterior to virtually see through its walls, viewing its interior objects and layout.[[48]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-48)[[49]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-49)

### Art

AR technology has helped disabled individuals create art by using [eye tracking](http://en.wikipedia.org/wiki/Eye_tracking) to translate a user's eye movements into drawings on a screen.[[50]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-50) An item such as a commemorative coin can be designed so that when scanned by an AR-enabled device it displays additional objects and layers of information that were not visible in a real world view of it.[[51]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-51)[[52]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-52)

### Commerce

[](http://en.wikipedia.org/wiki/File:ViewAR_BUTLERS_Screenshot.jpg)

[http://bits.wikimedia.org/static-1.21wmf9/skins/common/images/magnify-clip.png](http://en.wikipedia.org/wiki/File:ViewAR_BUTLERS_Screenshot.jpg)

ViewAR BUTLERS App - Placing furniture using AR

AR can enhance product previews such as allowing a customer to view what's inside a product's packaging without opening it.[[53]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-53) AR can also be used as an aid in selecting products from a catalog or through a kiosk. Scanned images of products can activate views of additional content such as customization options and additional images of the product in its use.[[54]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-54)[[55]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-55) AR is used to integrate print and video marketing. Printed marketing material can be designed with certain "trigger" images that, when scanned by an AR enabled device using image recognition, activate a video version of the promotional material.[[56]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-56)[[57]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-57)[[58]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-58)

### Education

Augmented reality applications can complement a standard curriculum. Text, graphics, video and audio can be superimposed into a student’s real time environment. Textbooks, flashcards and other educational reading material can contain embedded “[markers](http://en.wikipedia.org/wiki/Fiduciary_marker)” that, when scanned by an AR device, produce supplementary information to the student rendered in a multimedia format.[[59]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-59)[[60]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-60)[[61]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-61) Students can participate interactively with computer generated simulations of historical events, exploring and learning details of each significant area of the event site.[[62]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-62) AR can aide students in understanding chemistry by allowing them to visualize the spatial structure of a molecule and interact with a virtual model of it that appears, in a camera image, positioned at a marker held in their hand.[[63]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-63) Augmented reality technology also permits learning via remote collaboration, in which students and instructors not at the same physical location can share a common virtual learning environment populated by virtual objects and learning materials and interact with another within that setting.[[64]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-64)

### Everyday

30 years of Augmediated Reality in everyday life.Since the 1970s and early 1980s, Steve Mann has been developing technologies meant for everyday use i.e. "horizontal" across all applications rather than a specific "vertical" market. Examples include Mann's "EyeTap Digital Eye Glass", a general-purpose seeing aid that does dynamic-range management (HDR vision) and overlays, underlays, simultaneous augmentation and diminishment (e.g. diminishing the electric arc while looking at a welding torch).[[65]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-Quantigraphic_Father_of_AR-65)

### Industrial design

AR can help industrial designers experience a product's design and operation before completion. Volkswagen uses AR for comparing calculated and actual crash test imagery. AR can be used to visualize and modify a car body structure and engine layout. AR can also be used to compare digital mock-ups with physical mock-ups for efficiently finding discrepancies between them.

### Medical

Augmented Reality can provide the surgeon with information, which are otherwise hidden, such as showing the heartbeat rate, the blood pressure, the state of the patient’s organ, etc. In particular AR can be used to let the doctor look inside the patient by combining one source of images such as an [X-ray](http://en.wikipedia.org/wiki/X-ray) with another such as video. This helps the doctor to identify the problem with the patient in a more intuitive way than looking at only type of image data. This approach works in a similar as the technicians doing maintenance work.

Examples include a virtual [X-ray](http://en.wikipedia.org/wiki/X-ray) view based on prior [tomography](http://en.wikipedia.org/wiki/Tomography) or on real time images from [ultrasound](http://en.wikipedia.org/wiki/Ultrasound) and [confocal microscopy](http://en.wikipedia.org/wiki/Confocal_microscopy) probes[[69]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-69) or visualizing the position of a tumor in the video of an [endoscope](http://en.wikipedia.org/wiki/Endoscope).[[70]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-70) AR can enhance viewing a [fetus](http://en.wikipedia.org/wiki/Fetus) inside a mother's [womb](http://en.wikipedia.org/wiki/Womb).[[71]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-71) See also [Mixed reality](http://en.wikipedia.org/wiki/Mixed_reality).

### Military

In combat, AR can serve as a networked communication system that renders useful battlefield data onto a soldier's goggles in real time. From the soldier's viewpoint, people and various objects can be marked with special indicators to warn of potential dangers. Virtual maps and 360° view camera imaging can also be rendered to aid a soldier's navigation and battlefield perspective, and this can be transmitted to military leaders at a remote command center.[[72]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-72)

### Navigation

AR can augment the effectiveness of navigation devices. Information can be displayed on an automobile's windshield indicating destination directions and meter, weather, terrain, road conditions and traffic information as well as alerts to potential hazards in their path.

[](http://en.wikipedia.org/wiki/File:MediatedReality_on_iPhone2009_07_13_21_33_39.jpg)

Aboard maritime vessels, AR can allow bridge watch-standers to continuously monitor important information such as a ship's heading and speed while moving throughout the bridge or performing other tasks.[[76]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-76)

### Office workplace

AR can help facilitate collaboration among distributed team members in a work force via conferences with real and virtual participants. AR tasks can include brainstorming and discussion meetings utilizing common visualization via touch screen tables, interactive digital whiteboards, shared design spaces, and distributed control rooms.[[77]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-77)[[78]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-78)[[79]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-79)

### Sports and entertainment

AR has become common in sports telecasting. Sports and entertainment venues are provided with see-through and overlay augmentation through tracked camera feeds for enhanced viewing by the audience. Examples include the yellow "[first down](http://en.wikipedia.org/wiki/First_down)" line seen in television broadcasts of [American football](http://en.wikipedia.org/wiki/American_football) games showing the line the offensive team must cross to receive a first down. AR is also used in association with football and other sporting events to show commercial advertisements overlaid onto the view of the playing area. Sections of [rugby](http://en.wikipedia.org/wiki/Rugby_football) fields and [cricket](http://en.wikipedia.org/wiki/Cricket) pitches also display sponsored images. Swimming telecasts often add a line across the lanes to indicate the position of the current record holder as a race proceeds to allow viewers to compare the current race to the best performance. Other examples include hockey puck tracking and annotations of racing car performance and snooker ball trajectories. [[41]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-recentadvances-41)[[80]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-80)

AR can enhance concert and theater performances. For example, artists can allow listeners to augment their listening experience by adding their performance to that of other bands/groups of users.[[81]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-81)[[82]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-82)[[83]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-83)

The gaming industry has benefited a lot from the development of this technology. A number of games have been developed for prepared indoor environments. Early AR games also include AR air hockey, collaborative combat against virtual enemies, and an AR-enhanced pool games. A significant number of games incorporate AR in them and the introduction of the smartphone has made a bigger impact.[[84]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-84)[[85]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-85)

### Task support

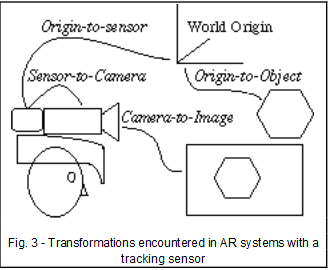
Complex tasks such as assembly, maintenance, and surgery can be simplified by inserting additional information into the field of view. For example, labels can be displayed on parts of a system to clarify operating instructions for a mechanic who is performing maintenance on the system.[[86]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-86)[[87]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-87) Assembly lines gain many benefits from the usage of AR. In addition to Boeing, BMW and Volkswagen are known for incorporating this technology in their assembly line to improve their manufacturing and assembly processes.[[88]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-88)[[89]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-89)[[90]](http://en.wikipedia.org/wiki/Augmented_reality#cite_note-90) Big machines are difficult to maintain because of the multiple layers or structures they have. With the use of AR the workers can complete their job in a much easier way because AR permits them to look through the machine as if it was with x-ray, pointing them to the problem right away.

### Tourism and sightseeing

Augmented reality applications can enhance a user's experience when traveling by providing real time informational displays regarding a location and its features, including comments made by previous visitors of the site. AR applications allow tourists to experience simulations of historical events, places and objects by rendering them into their current view of a landscape. AR applications can also present location information by audio, announcing features of interest at a particular site as they become visible to the user.

### Translation

AR systems can interpret foreign text on signs and menus and, in a user's augmented view, re-display the text in the user's language. Spoken words of a foreign language can be translated and displayed in a user's view as printed subtitles

**ARCHIETECTURE OF AUGMENTED REALITY** 

**Possible Applications**

Archeology

Architecture

Art

Commerce

Education

Industrial Design

Medical

Military

Navigation

Sports and Entertainment

Tourism and Travel

Translation

**REQUIREMENTS SPECIFICATIONS**

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

# H/W System Requirements: (Minimum h/w system requirements)

# Processor : Pentium –Core(i3)

Speed : 2.0 GHz

RAM : 2GB

Hard Disk : 20 GB

Key Board : Standard Windows Keyboard

Mouse : Two or Three Button Mouse

Monitor : SVGA

**Supporting devices (Phone specifications)**

OS : Android 2.0 and above

Inbuilt Camera : Yes

Camera Specifications : 3Megapixel Camera

# S/W System Requirements:(Minimum)

# Operating System : Windows95/98/2000/XP/vista/7/8/Ubuntu

Front End : Android XML

Database : Vuforia

Tool sets : NDK, QCAR

**Native development KIT [NDK]**

Libraries written in [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) and other languages can be compiled to [ARM](http://en.wikipedia.org/wiki/ARM_architecture) or [x86](http://en.wikipedia.org/wiki/X86) [native code](http://en.wikipedia.org/wiki/Native_code) and installed using the Android Native Development Kit. Native classes can be called from Java code running under the Dalvik VM using the System.loadLibrary call, which is part of the standard Android Java classes.

Complete applications can be [compiled](http://en.wikipedia.org/wiki/Compiler) and installed using traditional development tools. The ADB debugger gives a root shell under the Android Emulator which allows native [ARM code](http://en.wikipedia.org/wiki/ARM_architecture) or [x 86 codes](http://en.wikipedia.org/wiki/X86) to be uploaded and executed. ARM or x 86 codes can be compiled using [GCC](http://en.wikipedia.org/wiki/GNU_Compiler_Collection) on a standard PC. Running native code is complicated by Android's use of a non-standard C library (libc, known as [Bionic](http://en.wikipedia.org/wiki/Bionic_%28software%29)). The underlying graphics device is available as a [framebuffer](http://en.wikipedia.org/wiki/Framebuffer) at */dev/graphics/fb0*. The graphics library that Android uses to arbitrate and control access to this device is called the [Skia Graphics Library](http://en.wikipedia.org/wiki/Skia_Graphics_Engine) (SGL), and it has been released under an open source licence. Skia has backend for both [Win32](http://en.wikipedia.org/wiki/Windows_API) and [UNIX](http://en.wikipedia.org/wiki/Unix), allowing the development of cross-platform applications, and it is the graphics engine underlying the [Google Chrome](http://en.wikipedia.org/wiki/Google_Chrome) web browser.

Unlike Java application development based on the [Eclipse](http://en.wikipedia.org/wiki/Eclipse_%28software%29) IDE, the NDK is based on command-line tools and requires invoking them manually to build, deploy and debug the apps. Several third-party tools allow integrating the NDK into Eclipse and Visual Studio.

**QUALCOMM AUGMENTED REALITY [QCAR]**

The QCAR SDK 1.5 Beta1 and the QCAR Unity Extension introduces several changes to the 1.0 API, but in general retains the same functionality to detect and track trackables. New features like loading target datasets at runtime and several performance improvements to tracking robustness, and speed were made. Features of the QCAR SDK 1.5 Beta1 for Android include:

 Swappable datasets – allows a an app to dynamically load a previously created dataset based on some application level trigger

 Streamlined access to the video background texture to apply custom shaders for special visual effects,

 Detecting targets at steeper angles

 Improved robustness during tracking with less likelihood of losing targets

 Better pose estimate yielding more accurate tracking

 Less jitter in the pose calculation resulting in a more stable augmentation,

 Improved detection and tracking of Frame Markers, Faster QCAR system start-up times. This documentation and the QCAR SDK 1.5 Beta1, as well as the QCAR Unity Extension 1.5 Beta1 software packages are for Android only. If you have been developing applications with previous versions of the QCAR SDK or the QCAR Unity Extension you must apply several small changes to your existing project to make it compatible with the QCAR SDK 1.5 Beta1 and the QCAR Unity Extension. Please carefully review the corresponding section of this document for the full list of changes.

 Native SDK for Android – use this if you are developing with the native QCAR SDK

 Unity Extension for Android – use this if you use the QCAR Unity Extension

 DevGuide changes – this section summarizes the modifications to the DevGuide

 Public API – the appendix contains the updated public API documentation off-line. The beginning of the Native SDK and Unity Extension sections contains the updated Getting Started guide. It follows the same structure as published previously on the Developer Portal: https://ar.qualcomm.at/qdevnet/sdk. To ease your transition locating the subtle differences, we have highlighted the changes throughout the Getting Started sections. After the Getting Started guide you will find new and modified functionality explained. Deprecated API is marked with strikethrough. A step-by-step migration guide wraps up each of the first two sections to help transitioning your existing applications from QCAR 1.0 to the QCAR 1.5 Beta1.

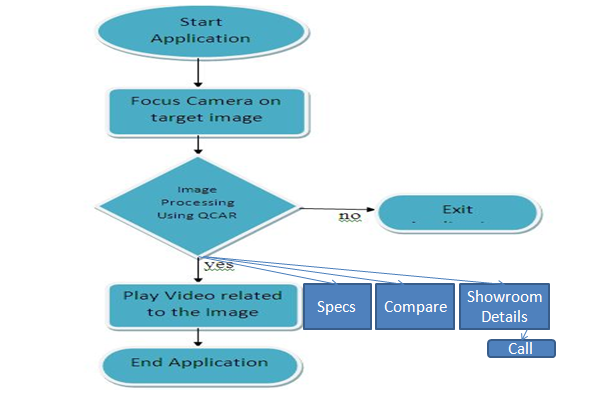
**CASE STUDY**

As there were no apps available in market for automobiles on the augmented reality and as the application would be easy to develop and can be made user friendly and easily made available to the customers.

And the few apps that are available in android gives you the site link along with the pictures where you need to catch the link and through the usage of internet u need to keep on browsing.

So we planned something interesting where we can just scan the logo present on the car and once the logo scanned will be matched we could easily obtain the triggered state where the video which has been made by stitching all the pictures for its description are available.

**DATA FLOW IMPLEMENTATION**

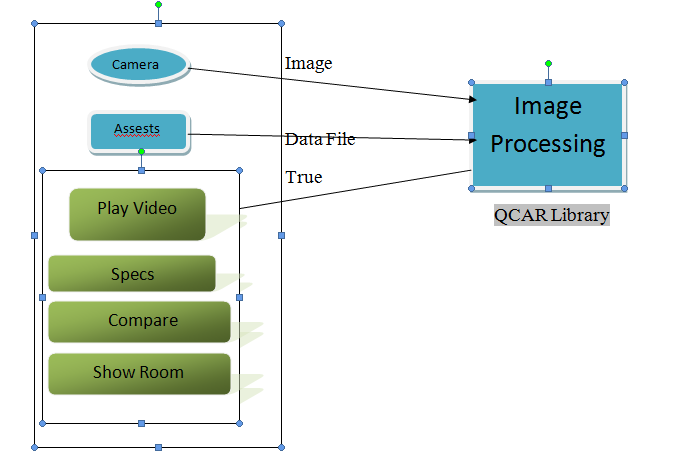


**Design**

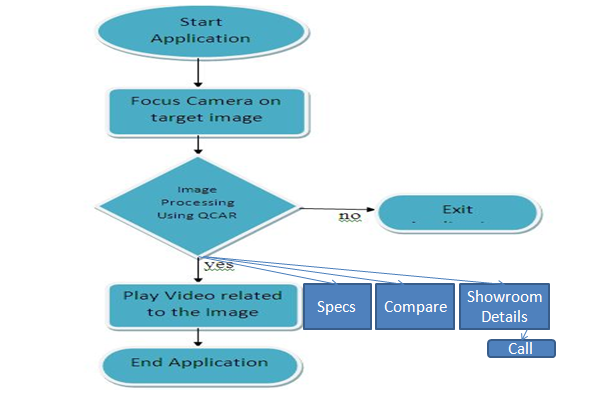
**Chapter-4**

**DESIGN**

**SYSTEM ARCHIETECTURE/TechnicAL Design**

****

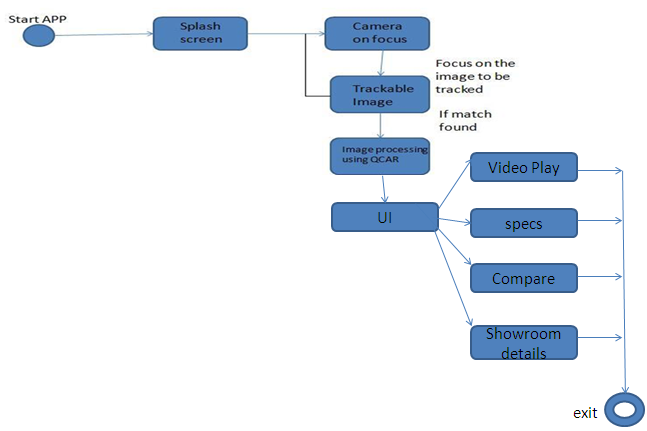
**workflow**



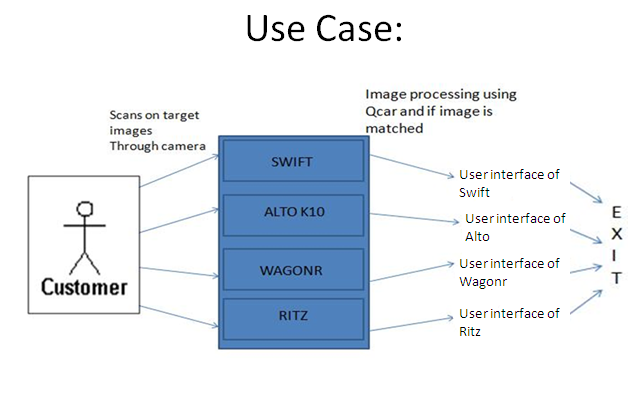
The project aims at developing an Android application that would have the capability to show:

* + - Information about a car
    - Technical specification
    - Interiors, exteriors, looks and feel
    - Price details and variants.

**State transition diagram**

****

**USE Case diagram**

****

**Implementation**

**Chapter-5**

**Implementation**

**Introduction**

Implementation is very simple as we need to just click on the button the app opens up with the camera on and once the camera finds the target it would provide the user interface such as provide the video play option, specification, showroom details and compare option.

**Implementation requirements**

**A**ndroid SDK

Android NDK

Eclipse IDE

Cygwin

QCAR

Android Smartphone with better camera quality.

**Platform selection**

**Android**

Android is an open source mobile operating system that combines and builds upon parts of many different open source projects. What does this mean to you as a developer? You have access to the source code of the platform that is running on the phone. This can help you better understand how interface controls and the various other pieces work. If you happen to find a bug, you can also submit a patch for the issue, though this is a more advanced practice. Google has also pulled together a large group of companies called the Open Handset Alliance that both contribute to and use the Android OS in their hardware devices. This means that there is industry-wide support for Google’s OS, promising wide adoption across well-known vendors.

**Android SDK(Eclipse)**

Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

Android is an open source operating system, created by Google specifically for use on mobile devices (cell phones and tablets)

Linux based.

Can be programmed in C/C++ but most app development is done in Java (Java access to C Libraries via JNI (Java Native Interface))

Supports Bluetooth, Wi-Fi, and 3G and 4G networking

**The Eclipse IDE**

Eclipse is a complex, multi-language, and extensible Integrated Development Environment (IDE). The learning curve can be steep, but the power of the environment can greatly increase your efficiency.

After opening Eclipse for the first time, select a workspace to save your project within. You will see an introduction screen with multiple icons. Select the “go to workbench” option, and you will be presented with the default project screen.

Assuming you have already installed the Eclipse ADT plugin, you will need to configure Eclipse for Android development by manually setting the filepath for the Android SDK. To do this, select **Eclipse Preferences** from the main tool bar, and then select Android from the dialogue box that appears. Update the “SDK Location” option to point to the directory where you installed the SDK. You should now have the IDE configured for Android development.

It is important to note that Eclipse uses something called “perspectives” to group commonly used tasks. Switching perspectives will switch out parts of the menu and toolbars, and will show and hide views related to them. Perspectives can be opened by clicking on the Open Perspective button or by choosing **Window Open Perspective**. Some perspectives that you will use frequently include Java, Debugging and DDMS.

**Introduction to QCAR SDK 1.5 Beta1 for Android**

**Overview**

The QCAR SDK 1.5 Beta1 and the QCAR Unity Extension introduces several changes to the

1.0 API, but in general retains the same functionality to detect and track trackables. New

features like loading target datasets at runtime and several performance improvements to

tracking robustness, and speed were made.

Features of the QCAR SDK 1.5 Beta1 for Android include:

Swappable datasets – allows a an app to dynamically load a previously created

dataset based on some application level trigger

Streamlined access to the video background texture to apply custom shaders for

special visual effects,

Detecting targets at steeper angles

Improved robustness during tracking with less likelihood of losing targets

Better pose estimate yielding more accurate tracking

Less jitter in the pose calculation resulting in a more stable augmentation,

Improved detection and tracking of Frame Markers, Faster QCAR system start-up

times.

This documentation and the QCAR SDK 1.5 Beta1, as well as the QCAR Unity

Extension 1.5 Beta1 software packages are for Android only.

If you have been developing applications with previous versions of the QCAR SDK or

the QCAR Unity Extension you must apply several small changes to your existing project to

make it compatible with the QCAR SDK 1.5 Beta1 and the QCAR Unity Extension. Please

carefully review the corresponding section of this document for the full list of changes.

**Document usage**

This document contains four main sections:

Native SDK for Android – use this if you are developing with the native QCAR SDK

Unity Extension for Android – use this if you use the QCAR Unity Extension

DevGuide changes – this section summarizes the modifications to the DevGuide

Public API – the appendix contains the updated public API documentation off-line

The beginning of the Native SDK and Unity Extension sections contains the updated Getting

Started guide. It follows the same structure as published previously on the Developer Portal:

https://ar.qualcomm.at/qdevnet/sdk. To ease your transition locating the subtle differences,

we have highlighted the changes throughout the Getting Started sections.

After the Getting Started guide you will find new and modified functionality explained.

Deprecated API is marked with strikethrough. A step-by-step migration guide wraps up each

of the first two sections to help transitioning your existing applications from QCAR 1.0 to the

QCAR 1.5 Beta1.

**Native SDK for Android**

**Getting Started**

If you are an Android developer and already have the Android SDK and NDK installed, then

please go directly to Step 2: Download and install the QCAR SDK. If you are new to Android

software development, you will find it very useful to start at Step 1 below.

1. Setting-up the development environment

2. Installing the QCAR SDK

3. Compiling & Running a QCAR sample app

If you have used the QCAR SDK in the Beta phase you might want to check the section on

application publishing. Note that several changes have been applied to the environment setup

and some API features that you should revisit.

**Using the On-Line Documentation**

The first step for both newcomers and experienced AR developers is to follow the Getting

Started guide above and run a QCAR sample app. After that, depending on your skill and

comfort level with AR, you may go directly to one of the following options:

* If you are familiar with the concepts of vision based augmented reality, then you can
* look at the Developer Guide to learn more about QCAR SDK features.
* If you are a "No text - unless it starts with //" type of person, then the API Reference

is the right place for you!

**Step 1: Setting up the Development Environment**

**Supported Development Platforms**

The QCAR SDK supports Android OS 2.1 and above.

The recommended development environment is Microsoft Windows 7 32/64bit or Windows

XP.

The components to build the actual code (JDK, Eclipse+ADT and gcc) are available across

multiple platforms. While building on Linux Ubuntu and Mac OS X environments is possible,

we will be unable to support those platforms with detailed documentation. However, we have

included hints and notes to platform specific issues that will help you set up your

development environment on Linux Ubuntu 11.10 or Mac OS X 10.6 (Snow Leopard). This

set-up guide has been written for the Win7 32/64bit platform with special notes for other

operating systems.

If you already have both the Android SDK and the NDK setup, then please go directly

to Step 2: Installing the QCAR SDK.

**Set-up the Android Development Environment**

The QCAR SDK requires both the Android SDK and the NDK. The Android NDK is an

extension to the Android SDK that lets Android developers build performance-critical parts of

their applications in native code. SDK and NDK communicate over the Java-Native-Interface

(JNI).

To set-up the development environment please install the following components in the order

they appear below:

1. JDK

2. Eclipse IDE

3. Android SDK Downloader

4. Android ADT

5. Android SDK platform support

6. Cygwin Environment

7. Android NDK

We recommend to use the following versions of the above listed tools with the QCAR 1.5

Beta1 SDK:

* Java SE Development Kit: use version Java SE 7u1
* Eclipse: use latest version
* Android SDK Downloader: use version Android SDK Tools revision 15
* Android ADT: choose the latest version that is for SDK tools rev 15
* Android SDK Platform Support: use Android SDK Platform-tools revision 9
* Cygwin Environment: use latest version 1.7.9-1.
* Android NDK: use version Android NDK 7

***2.1.2.1 JDK***

Download the **Java SE Development Kit** (JDK) from the following site:

http://www.oracle.com/technetwork/java/javase/downloads/

Click the 'Download' button from the JDK section of the 'Java Platform, Standard Edition'

table.

Install the JDK environment with default settings.

Detailed installation instructions and system requirements can be found at:

http://www.oracle.com/technetwork/java/javase/index-137561.html

***2.1.2.2 Eclipse IDE***

In this step you will install the Eclipse IDE. Download the latest version of **Eclipse IDE for**

**Java Developers** from:

http://www.eclipse.org/downloads/

Unpack the downloaded ZIP package and copy the contents of the archive starting with the

subdirectory 'eclipse' to your program directory path in C:\Program Files\eclipse.

You may also create a shortcut to eclipse.exe on your desktop or start menu.

Start eclipse.exe

The very first time that Eclipse is started, the IDE will ask you to assign storage space for

your workspace. This directory will contain ONLY IDE specific settings and information. This

is not your application development workspace (despite the notice in this window). This type

of information is typically stored in the user's home directory, which is the default value here:

C:\Users\USERNAME\workspace. Check "Use this as the default and do not ask again".

***2.1.2.3 Android SDK Downloader***

The Android SDK is distributed through an SDK starter package containing the SDK Tools.

Download the starter package from:

http://developer.android.com/sdk/index.html

Unzip the archive and copy the contents into a directory,

Ex. :\Development\Android\android-sdk-windows\.

Throughout the Get Started guide we will refer to the base directory of your development

environment as

<DEVELOPMENT\_ROOT>= C:\Development\Android

Add the tools\ directory to your Windows path. Right-click on Computer on the desktop

and select Properties. Use the Advanced system settings to open the System

Properties window and select Environment Variables on the Advanced tab. Look for

variable Path in the System variables window. After pressing Edit, scroll to the end of

Variable value: and add the full path to the tools\ directory to the end of the path,

separated by a semicolon from the one before. For example:

;C:\Development\Android\android-sdk-windows\tools\

**Troubleshooting**

For troubleshooting Android related issues and for more detailed instructions on the Android

SDK set-up, please refer to the following:

* Quick Start Android SDK set-up
* Installing the Android SDK

***2.1.2.4 Android ADT***

Android Development Tools (ADT) is a powerful extension to Eclipse that connects it with

the Android SDK and helps with app development. This package is installed from within

Eclipse. Select Help->Install New Software... . Add the url

https://dl-ssl.google.com/android/eclipse/

into Work with field. Eclipse will ask you to provide an arbitrary name for the update site.

After a short while Developer Tools appears in the field. At the very minimum, please

select Android Development Tools and Android DDMS - which adds debugging support -

from the list, and then click on Next-> .

Page 9 of 73

*Eclipse component selection*

After reviewing this selection and accepting the license terms, the downloader fetches the

files and puts them in your Eclipse directory. After accepting the certificate and an automatic

restart of Eclipse, the installation of the Android ADT is complete.

***2.1.2.5 Android SDK Platform Support***

To develop for Android, support for the appropriate Android platform must be installed. The

**Android SDK Manager** is used to install additional components and support for different

platforms. Select in Eclipse the menu Window->Android SDK Manager. In case the Android

SDK location was not setup correctly within Eclipse, go to Window->Preferences-

>Android and set the SDK location field to the root of your SDK install directory.

In the Android SDK Manager window, sort by API level, click Deselect All, and check

the following boxes:

From Tools:

* Android SDK Platform-tools
* From Android 4.0 (API 14):
* Documentation for Android SDK
* From Android 2.2 (API 8):
* Samples for SDK (optional)
* From Android 2.1 (API 7):
* SDK Platform
* From Extras:
* Google USB Driver package (not compatible with Linux)

*Android SDK component selection*

To install the selected options you must hit Install 5 packages and accept all licenses on

the next window. Use Accept All as a shortcut and hit Install.

*Android component license acceptance*

Finally, add the platform-tools\ directory to your Windows path:

;C:\Development\Android\android-sdk-windows\platform-tools\

***2.1.2.6 Cygwin Environment***

A GNU compiler is required to compile dynamic applications as shared libraries for the

Android NDK. Android make files are designed to run with gcc4. On Windows, a convenient

way to have the complete environment prepared for this, is to install Cygwin.

Cygwin uses an installer helper to manage the installation process. Go to

http://www.cygwin.com/setup.exe and select "Install from Internet" when prompted at

"Choose a Download Source" in the installer. We recommend not changing the Root

Directory in the next window, and leaving it at "C:\cygwin". The "Local Package

Directory" holds the downloaded packages. You may want to keep them with the

downloaded Setup.exe in the same directory so as to have a Cygwin installer directory.

Choose a download site with a known fast connection near you.

When the package information is downloaded you will see a hierarchical browser to select

packages. Select the following package from the hierarchy for download:

All ->Devel -> "make: The GNU version of the 'make' utility"

Select the word "skip" to change it to the actual version number, which is currently 3.81-2.

Finish the installation by clicking next.

Your Cygwin environment is fully set-up to work with the QCAR SDK. If you have other

similar environments installed, make sure to set your Windows path variable to point to

"C:\cygwin\bin" so that bash uses this version of GNU's make.exe.

***2.1.2.7 Android NDK***

The Android NDK is an extension to the Android SDK that lets Android developers build

performance-critical parts of their applications in native code. Download the NDK package

from:

http://developer.android.com/sdk/ndk/index.html

Unzip the archive and copy the contents into a directory. To be consistent with our previous

setup we recommend that you put the contents in "C:\Development\Android\androidndk-

r7\". Thus Android SDK and Android NDK share the same parent directory. We will

later add the QCAR-SDK and your project files here.

NDK requires the above directory to be added to the Windows path! To do this right-click on

'My Computer' on the desktop and select properties. On the 'Advanced' tabs select

'Environment Variables' and look for Variable 'Path' in the 'System variables' window.

After pressing 'Edit', scroll to the end of 'Variable value:' and add the full path to the

directory to the end of the path, separated via semicolon from the one before.

In the above example, you would have to add:

;C:\Development\Android\android-ndk-r7\

To test your installation you can compile any of the NDK sample applications. Using a

Cygwin bash shell, navigate to the root directory of any demo application (e.g. for the 'sanangeles'

sample app with out installation path above):

cd /cygdrive/c/Development/Android/android-ndk-r7/samples/san-angeles

ndk-build

The compiler should produce a dynamically linked library libsanangeles.so and write it to

/libs/armeabi within the application directory. NDK includes support for different

architectures so you might find different subdirectories in /libs.

Now your development environment is ready to host QCAR SDK related content.

**2.1.3 Step 2: Installing the QCAR SDK**

**Clean Installation**

QCAR SDK is distributed using installers for the following platforms: Windows, Mac OS X,

and Linux.

To start developing with the QCAR SDK you will need to,

* Download & Install QCAR-SDK under <DEVELOPMENT\_ROOT>.
* Adjust QCAR Environment settings in Eclipse.

After accepting the license agreements the installer creates a directory structure in your

Android development environment. This structure will ensure that sample apps can be easily

built and deployed using the Android NDK and the Eclipse Java developer environment.

**Upgrading from a Previous Version**

When installing an updated version of the QCAR SDK, we suggest that you install it under a

new directory tree, just as the installer recommends. To compile the new set of sample

applications it is required to adjust the Eclipse workspace variable setting. Lastly you will

need to move your projects under the new <DEVELOPMENT\_ROOT> (see below).

**Resulting Directory Structure**

To streamline development we have designed a directory structure that keeps the QCARSDK

and your applications in separate trees. This will ensure easier updates to the SDK,

leaving your application source-trees untouched.

In the previous steps we used a starting directory for the SDK and the NDK installation that

we called:

<DEVELOPMENT\_ROOT>= C:\Development\Android

The downloaded installer creates a directory structure that will integrate into your Android

development environment. Following the recommended installation location the development

environment should finally result in the structure below. Here xx-yy-zz and xx.yy.zz stand for

the version number of the QCAR SDK.

<DEVELOPMENT\_ROOT>\

android-ndk-r7\

android-sdk-windows\

qcar-android-xx-yy-zz\

build\ QUALCOMM Augmented Reality SDK

licenses\ License Agreements

samples\ Sample applications with full source code

assets\ Additional assets for the QCAR SDK

readme.txt Starting read-me document

***2.1.3.1 Install the QCAR SDK***

**Download**

QCAR SDK is distributed through the Qualcomm Augmented Reality Developer Platform site.

Follow the instructions below to access the installer.

Although we distribute installers for the platforms listed below, support for QCAR

SDK development is limited to the Win 7 32/64 bit platform.

**Windows**

Although we recommend that you develop on Windows 7 32/64-bit or Windows XP 32-bit

environments, the QCAR SDK has also been successfully run on Windows XP 64-bit.

However, we currently do not offer specific support for this platform.

1. Download installer EXE-file from the Downloads page.

2. Run installer

***2.1.3.2 Set the QCAR Environment Variable***

Our recommended directory structure allows for SDK upgrades independent of application

development. Your future AR projects will be in the recommended structure under

<DEVELOPMENT\_ROOT>. One workspace variable must be set in Eclipse so that is aware of

this hierarchy:

In Eclipse, go to Window->Preferences. Navigate to Java->Build Path->Classpath

Variables using the hierarchy browser. Create a new variable by selecting New....

Add

QCAR\_SDK\_ROOT

into the Name: field and navigate using Folder... to the <DEVELOPMENT\_ROOT>\qcarandroid-

xx-yy-zz directory we defined in the QCAR SDK Setup section. In our example

above the variable value is

C:\Development\Android\qcar-android-xx-yy-zz

where xx-yy-zz denotes the current QCAR SDK version number.

This setting is essential as the classpath settings in the sample files use this reference to

include common shared JAR files.

***2.1.3.3 Prepare Test Device for Development***

**Developer Settings on the Device**

Android devices require special settings for development. In this step we will:

* Allow apps to be installed from unknown sources
* Enable USB debugging

On the device, go to Settings->Applications and choose "Unknown sources" as shown

below. Accept the warning shown on the right. This setting allows the direct installation of

unsigned APKs from within Eclipse.

*Allowing unknown sources for software installation*

Go to the Development dialog on the above screen and choose the two settings as shown

below:

*USB debugging is mandatory. The stay awake setting will help with development.*

**Install the USB Driver**

Connect your device to the development PC via the USB cable.

On initial connection, Windows will recognize the new device and attempt to locate a

compatible a driver. The Android SDK already includes some USB drivers - others will have

to be obtained directly from the device manufacturer. SDK pre-packaged drivers can be

located in the following directory:

<DEVELOPMENT\_ROOT>\android-sdk-windows\extras\google\usb\_driver

When the device installation finishes you are ready to use your test device.

On some machines the USB driver installation will ask you to reboot the machine. You can

skip this step and the device should be accessible without a reboot.

**2.1.4 Step 3: Compiling & Running a QCAR Sample App**

We are now going to build a sample application included in the QCAR SDK package. The

ImageTargets application is a good place to start learning about the SDK as it shows

detection and tracking of natural features using common images. This section will show you

how to build the native C++ source files with the NDK package of the Android SDK and then

use Eclipse to build the Java sources and create the APK package that can be deployed to

the device.

***Create the Android APK***

In the previous step, we created the shared object for the ImageTargets sample app. We

must now compile and build the Android application APK using the Eclipse IDE environment.

The sample applications have several Java classes to create the bootloader, provide GUI

functionality, read the orientation sensor of the device, allow user interaction, and provide

video background for the app.

**Building the ImageTargets application**

Create a new project in Eclipse by selecting File->New->Project.... Choose Android-

>Android Project from the Wizard selection and click next. Select 'Create project from

existing source' and browse to the <DEVELOPMENT\_ROOT>\qcar-android-xx-yyzz\

samples\ImageTargets directory. Click Finish.

*Eclipse "New Android Project" panel*

The standard installation of Eclipse has automatic compilation turned on. The very first time

you compile the app, you may receive some error messages from Eclipse. Project-

>Clean...->Clean All Projects should fix these errors as Eclipse has to build some

standard directories. This also creates the APK package for deployment.

***2.1.4.3 Run the Sample Application***

**Print the Image Target**

Print all image targets in <DEVELOPMENT\_ROOT>\qcar-android-xx-yyzz\

samples\ImageTargets\media from any of the formats onto a US Letter or A4 sized

paper with the page scaling 'none' option, keeping the original aspect ratio of the image

intact.

"chips" Image Target "stones" Image Target

“tarmac” Image Target

**Deploy & Run the Application**

With the device connected, select the "ImageTargets" project from the Package Explorer in

Eclipse and choose Run->Run As-> Android Application.

Eclipse automatically installs it to a connected Android device using ADB, and starts running

it on the device.

*You have successfully deployed your first application with the QCAR-SDK!*

After a splash screen the live camera image is shown. Use your device to look at the printed

target. You should see a textured teapot centered on top of the target, registered to the

plane.

Next, press the menu button, and select “Switch to Tarmac dataset”, and then hold your

device up against the “tarmac” printed target and you will see the teapot, only now it is

orange.

The ImageTargets app shows how the SDK can detect and track a single image from a predefined

set of images. The app also shows how to switch between datasets without too

much effort, as well as how to deal with camera settings, such as autofocus. With a small

code change, it can also be compiled to detect and track multiple images simultaneously.

Check the Tips & Tricks section in the Developer Guide for more information on this.

***QCAR::ImageTracker***

The ImageTracker class – derived from the QCAR::Tracker class – tracks Image

Targets and Multi Targets contained in a QCAR 1.5 Beta1 compatible target dataset. The

ImageTracker class provides methods for creating, activating and deactivating datasets.

All dataset operations shall be called only while the ImageTracker is not active. During the

execution of the QCAR::UpdateCallback this is guaranteed.

While multiple datasets can be loaded as explained in the DataSet class description, only

one dataset can be active at any given time. Before using another dataset you must

deactivate the currently active dataset, otherwise activateDatSet() returns FALSE. This

also happens if the dataset to be activated was already active.

**Testing**

Chapter 6

Testing

Introduction

 Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well. Correctness testing and reliability testing are two major areas of testing.

Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

Unlike most physical systems, most of the defects in software are design errors, not manufacturing defects. Software does not suffer from corrosion, wear-and-tear -- generally it will not change until upgrades, or until obsolescence. So once the software is shipped, the design defects -- or bugs -- will be buried in and remain latent until activation. Because software and any digital systems are not continuous, testing boundary values are not sufficient to guarantee correctness. All the possible values need to be tested and verified.

A further complication has to do with the dynamic nature of programs. If a failure occurs during preliminary testing and the code is changed, the software may now work for a test case that it didn't work for previously. But its behavior on pre-error test cases that it passed before can no longer be guaranteed. To account for this possibility, testing should be restarted. The expense of doing this is often prohibitive.

Levels of Testing

* Unit Testing

Unit Testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module but is more commonly an individual function or procedure.

The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.

A manual approach to unit testing may employ a step-by-step instructional document. Nevertheless, the objective in unit testing is to isolate a unit and validate its correctness. Automation is efficient for achieving this, and enables the many benefits listed in this article.

Creating Unit tests

You can write the JUnit tests manually but Eclipse supports the creation of JUnit tests via wizards.For example to create a JUnit test or a test class for an existing class, right-click on your new class, select this class in the *Package Explorer* *view*, right-click on it and select *New* → *JUnit Test Case*.

Alternatively you can also use the JUnit wizards available under File → New → Other... → Java → JUnit.

**Running JUnit tests**

To run a test, select the class which contains the tests, right-click on it and select Run-as → JUnit Test. This starts JUnit and executes all test methods in this class.

Eclipse provides the **Alt**+**Shift**+**X,** **,T** shortcut to run the test in the selected class. If you position the cursor on one method name, this shortcut runs only the selected test method.

To see the result of an JUnit test, Eclipse uses the JUnit view which shows the results of the tests. You can also select individual unit test in this view, right-click them and select Run to execute them again.

* Integration Testing

Integration Testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using Black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly. Some different types of integration testing are big bang, top-down, and bottom-up.

The Maven Failsafe Plugin is designed to run integration *tests*. The Maven lifecycle has four phases for running integration tests:

* Pre-integration-test for setting up the integration test environment.
* Integration-test for running the integration tests.
* Post-integration-test for tearing down the integration test environment.
* Verify for checking the results of the integration tests.

The Failsafe Plugin is used during the integration-test and verify phases of the build lifecycle to execute the integration tests of an application. The Failsafe Plugin will not fail the build during the integration-testphase thus enabling the post-integration-test phase to execute.

NOTE: when running integration tests, you should invoke maven with the (shorter to type too)

mvn verify

rather than trying to invoke the integration-test phase directly, as otherwise the post-integration-testphase will not be executed.

The Failsafe Plugin generates reports in 2 different file formats:

* Plain text files (\*.txt)
* XML files (\*.xml)

By default, these files are generated at ${basedir}/target/failsafe-reports.

The Failsafe Plugin has only 2 goals:

* [failsafe: integration-test](http://maven.apache.org/surefire/maven-failsafe-plugin/integration-test-mojo.html) runs the integration tests of an application.
* [failsafe: verify](http://maven.apache.org/surefire/maven-failsafe-plugin/verify-mojo.html) verifies that the integration tests of an application passed.

Running a Single Test:

During development, you may run a single test class repeatedly. To run this through Maven, set the it.test property to a specific test case.

mvn -Dit.test=ITCircle verify

* System testing

System testing of software or hardware is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

* Validation

Validation is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle.

Validation ensures that "you built the right thing". Validation confirms that the product, as provided, will fulfill its intended use.

Validation in Eclipse:

When saving a java source file that resides in a project, eclim will update that source file in Eclipse and will report any validation errors found. Any errors will be placed in the current window's location list (:help location-list) and the corresponding lines in the source file will be marked via Vim's :sign functionality with '>>' markers in the left margin.

Automatic validation of java source files can be disabled via the **g:EclimJavaValidate** variable. If you choose to disable automatic validation, you can still use the **:Validate** command to manually validate the current file.

* Output

Output is the term denoting either an exit or changes which exit a system and which activate/modify a process. It is an abstract concept, used in the modeling, system(s) design and system(s) exploitation.

Output in Eclipse:

Java standard output will show in the console in eclipse.You can show the view by the following steps:

Open Window > Show View > Console.

Rather than trying to output to the console, Log will output to LogCat which you can find in Eclipse by going to: Window->Show View->Other…->Android->LogCat.

User Acceptance

Acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests.

Software developers often distinguish acceptance testing by the system provider from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership. In the case of software, acceptance testing performed by the customer is known as user acceptance testing (UAT), end-user testing, site (acceptance) testing, or field (acceptance) testing.

**User Acceptance Testing** (UAT) is a process to obtain confirmation that a system meets mutually agreed-upon requirements. ASubject Matter Expert (SME), preferably the owner or client of the object under test, provides such confirmation after trial or review. Insoftware development, UAT is one of the final stages of a project and often occurs before a client or customer accepts the new system.

Test designers draw up formal tests and devise a range of severity levels. Ideally, a system's user acceptance tests and its formal integration and system [test cases](http://en.wikipedia.org/wiki/Test_case) should not be drafted by the same person. The UAT acts as a final verification of the required business function and proper functioning of the system, emulating real-world usage conditions on behalf of the paying client or a specific large customer.

User tests, which are usually performed by clients or end-users, do not normally focus on identifying simple problems such as spelling errors and cosmetic problems, nor [showstopper](http://en.wikipedia.org/wiki/Software_bug) defects, such as software crashes; testers and developers previously identify and fix these issues during earlier [unit testing](http://en.wikipedia.org/wiki/Unit_testing), [integration testing](http://en.wikipedia.org/wiki/Integration_testing), and system testing phases.

The results of these tests give confidence to the clients as to how the system will perform in production. There may also be legal or contractual requirements for acceptance of the system.

User Acceptance Testing in Eclipse:

This can be done by using UI testing tools such as Jubula Functional Testing tool, Q7 tool etc.,

* GUI Testing

**Graphical User Interface Testing** is the process of testing a product's graphical user interface to ensure it meets its written specifications. This is normally done through the use of a variety of test cases.

GUI testing in Eclipse:

This can be done by using GUI testing tools such as Squish tool,SWTBot tool, Xored Q7 tool etc,.

**Screen shots**

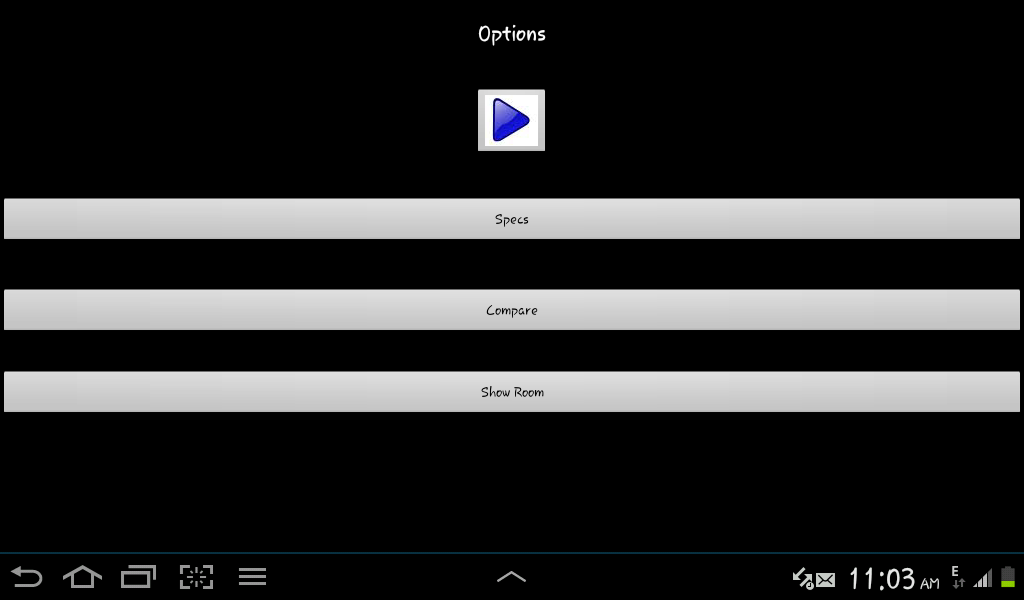
Chapter 7

**Screen shots**

This application provides the options after scanning such as play video , specification details, compare, showroom details and also the call facility

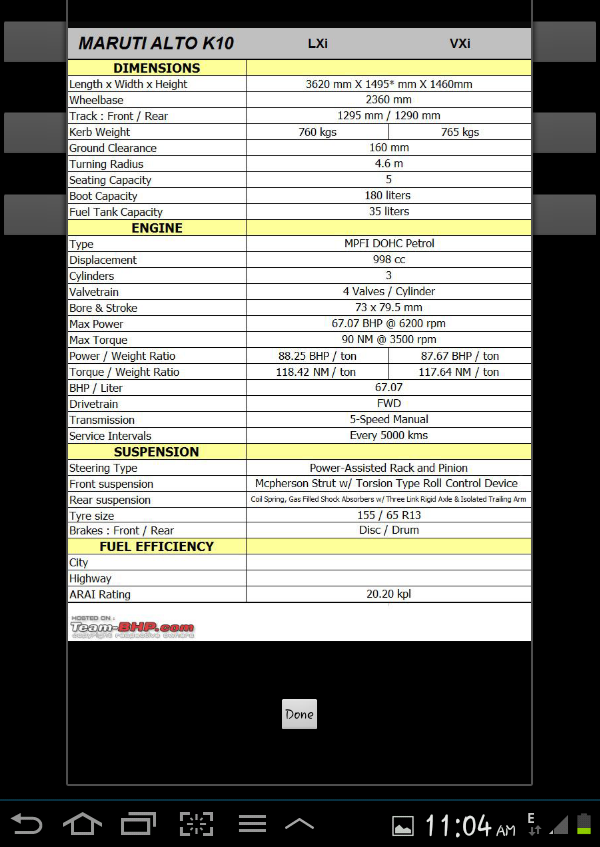
showroom

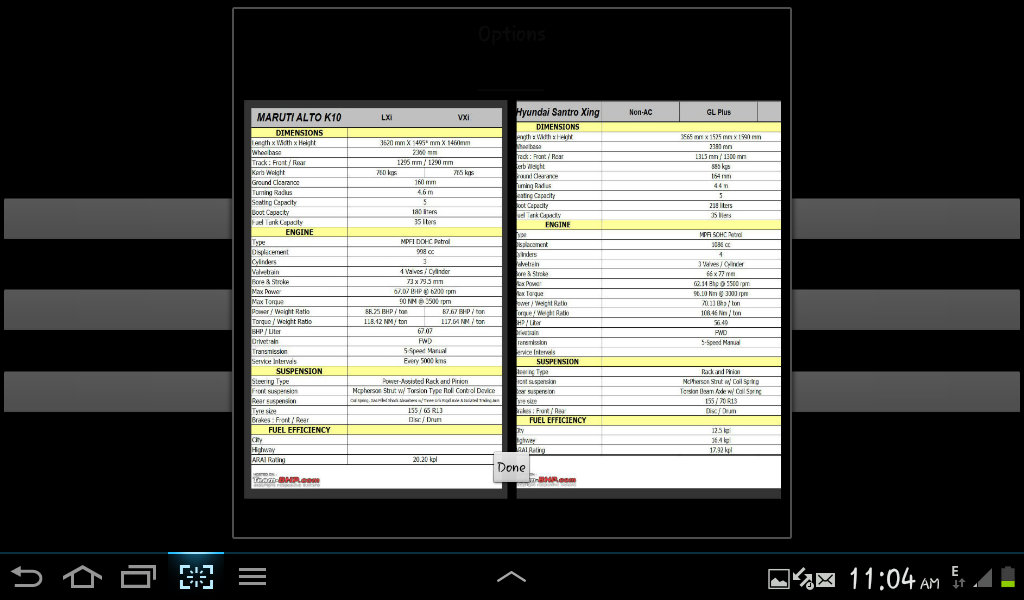


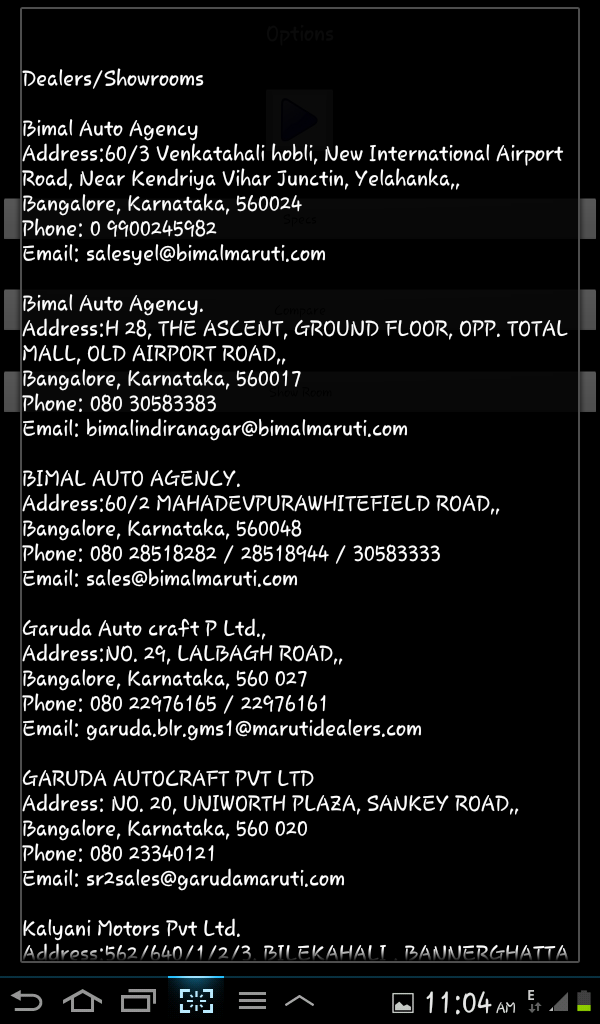


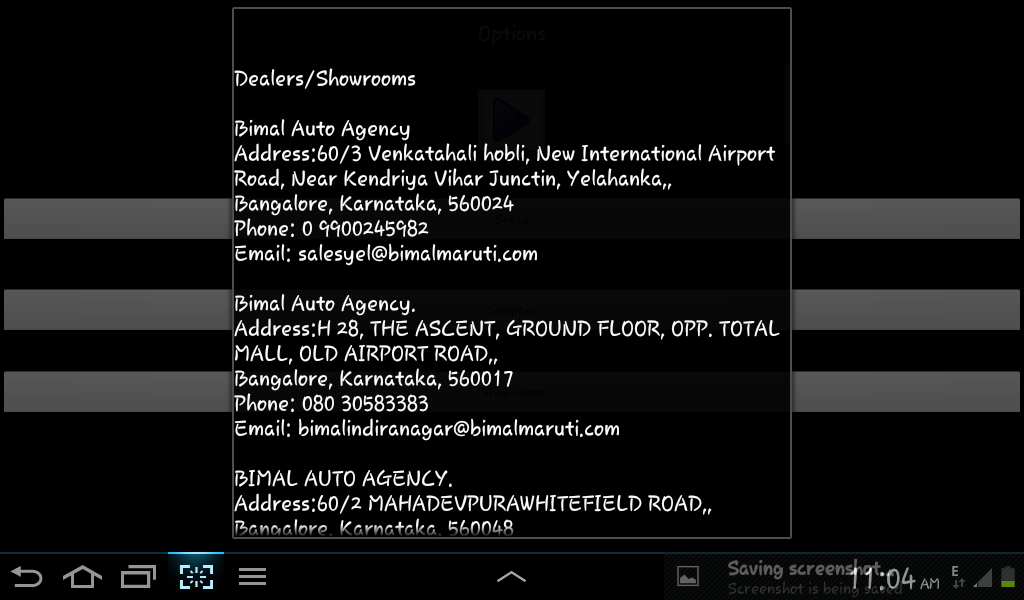


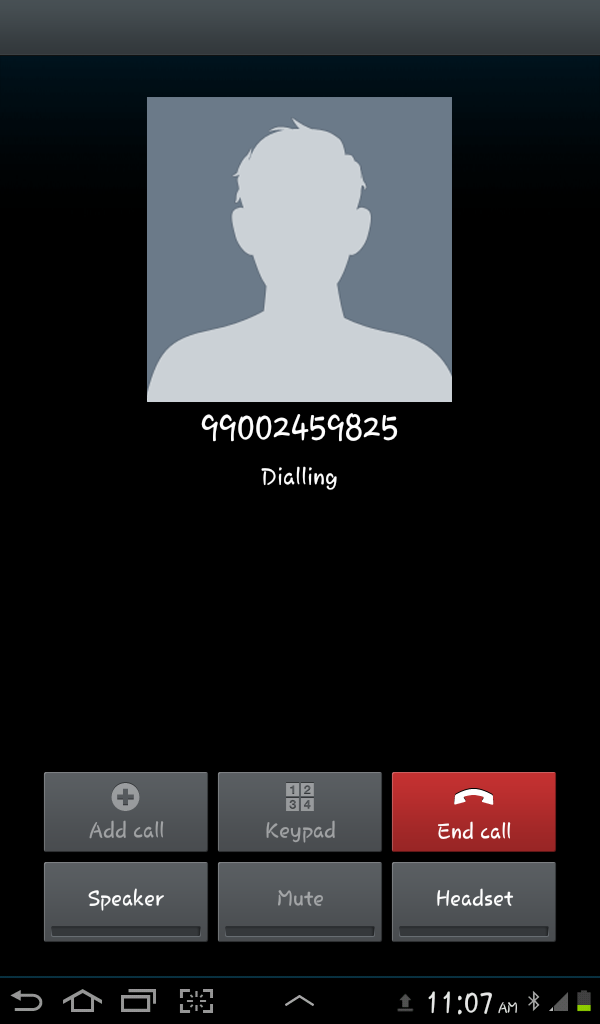
This is the screenshot of video playing which gives the complete view and price details and colour options











**CONCLUSION**

Chapter 8

Conclusion

It is a free application, which is easy to use, universally available, simple, thorough Application that helps making choices relating to automobile decisions easier and pleasing to the eye.

The success of a system like the one proposed here would depend on the acceptance of customers who gets the complete description of automobile without internet.

This application have the capability to show:

* + - Information about a car
    - Technical specification
    - Interiors, exteriors, looks and feel
    - Price details and variants

**Bibliography**

Bibliography

* [www.devepoler.vuforia.com](http://www.devepoler.vuforia.com)
* [www.qualcomm.co.in](http://www.qualcomm.co.in)
* [www.maruthisuzuki.com](http://www.maruthisuzuki.com)
* [www.developer.augmentedreality.com](http://www.developer.augmentedreality.com)