ECE 574: Advanced Software Techniques in Engineering Applications

Instructor: Dr. Adnan Shaout

Student Name: Srinivas Sambaraju

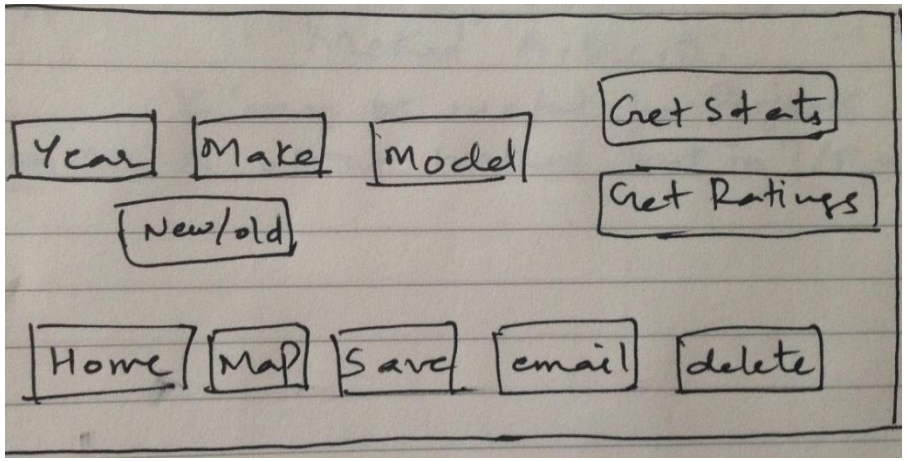
Date: Nov 20, 2014

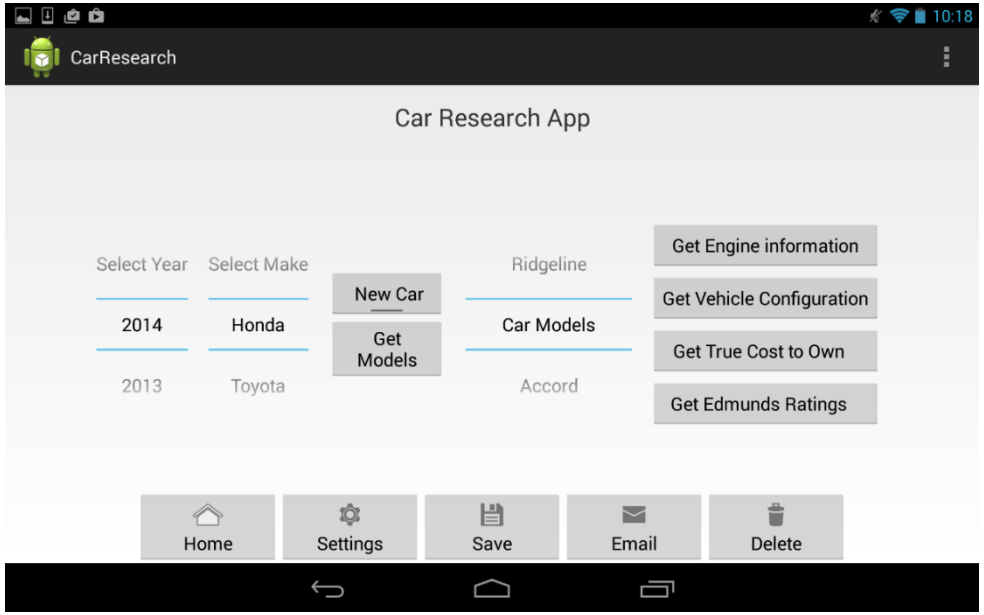
Project Presentation

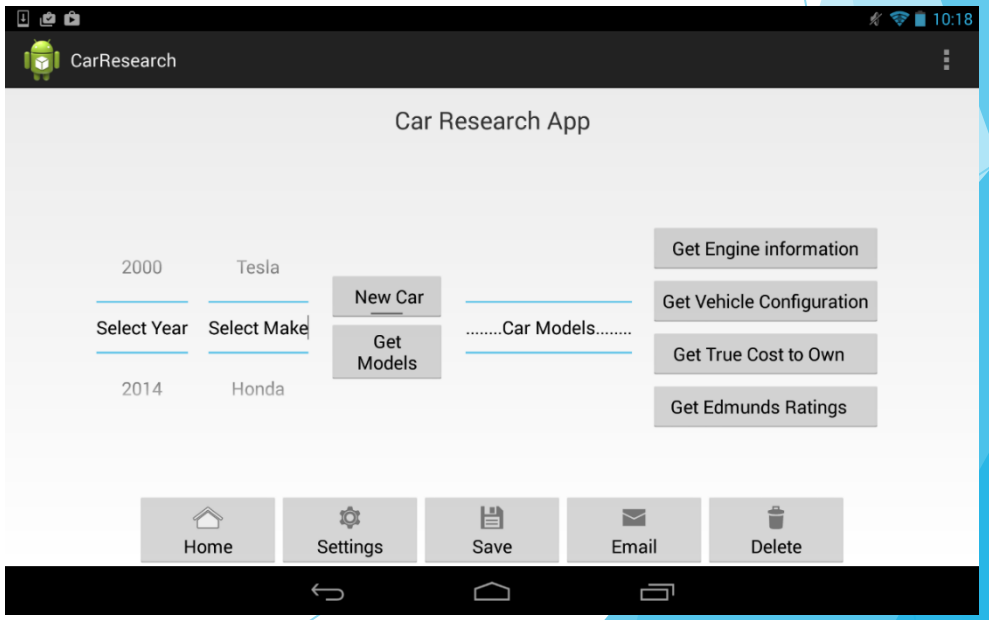
Agenda

- Live demo of the Car Research Android App
- Software process used
  - what, why, how
- Describe the software process
- Describe how the process was applied in various stages
  - Concept, Requirements, Review, Design, Development, Testing, Maintenance.
- Describe steps to develop App
  - Edmunds API, technical details, and flow chart etc..
- Observations and learning from the project.
- Quick summary

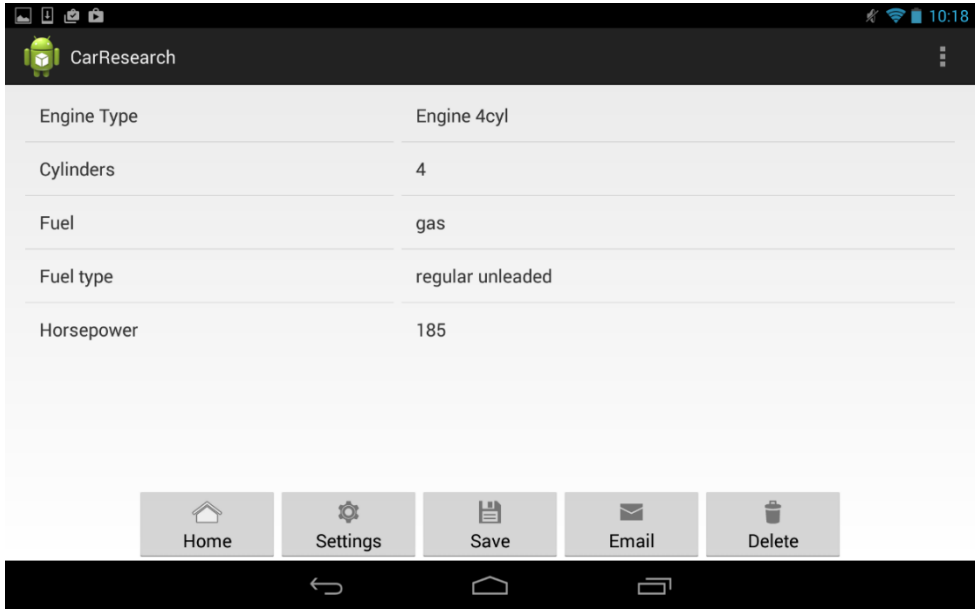
App Screen shots – Preliminary and Final

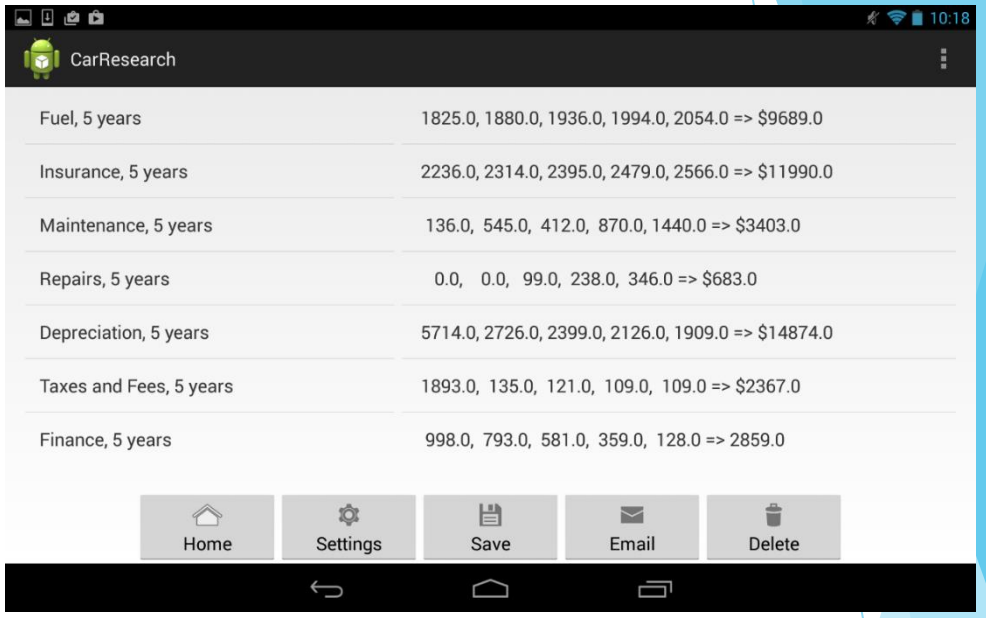


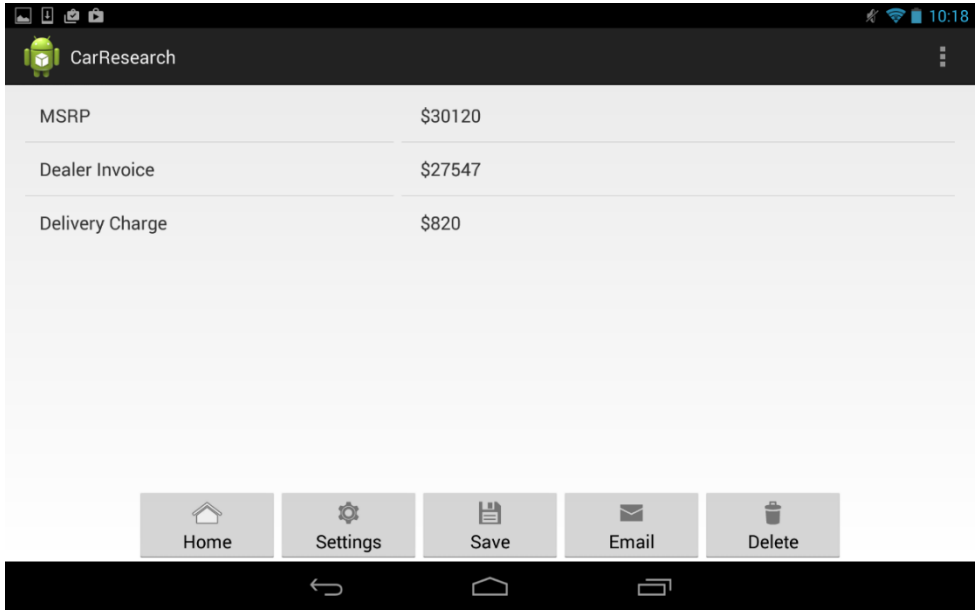


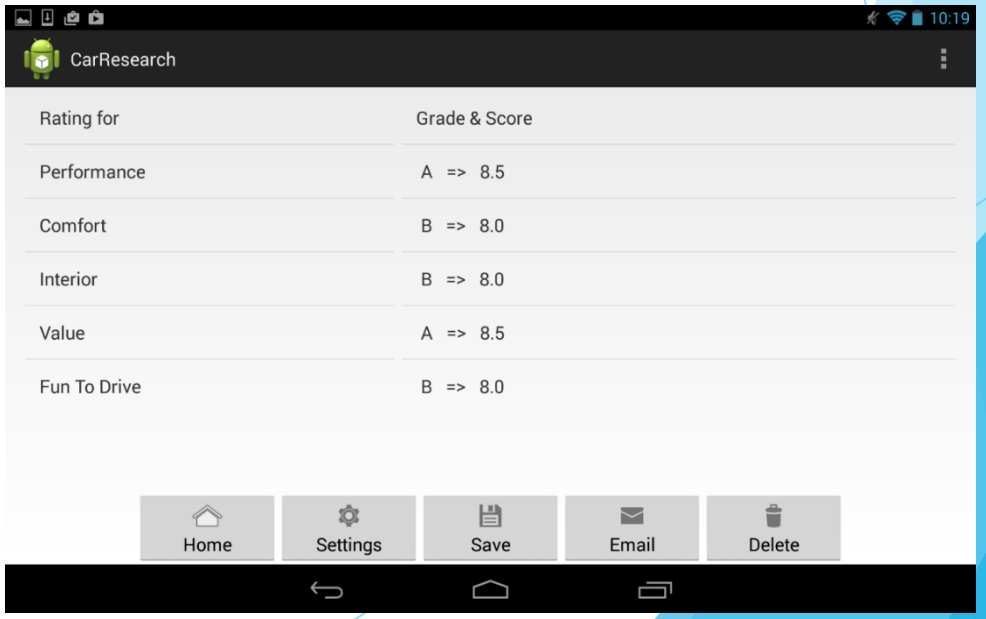


App Screen shots…*continued..*









Software Process

- What process is used?
  - XP (eXtreme Programming) Agile software development process was followed.
- Why it was used?
  - This project was about developing an Android App in a short period of time with a single developer.
  - For this, XP is very suitable because it is an iterative process that proceeds in steps where each step could include design, coding, testing and updated requirements.
  - The project can be completely visualized with the GUI itself and obtaining relevant information from the App determined the success.
  - As a developer I could make decisions, run a quick iteration and make corrections.
- Did the process help in each phase?
  - The process was definitely helpful because development proceeded in many iterations with specific targets for each iteration and within a specific time.
  - Only certain part of the current phase of the project has to be completed to advance to the next phase and the remaining work was covered in the next iteration.

Definition of Agile Process [1]

An iterative and incremental (evolutionary) approach to software development
which is performed in a highly collaborative manner
by self-organizing teams within an effective governance framework
with "just enough" ceremony
that produces high quality solutions
in a cost effective and timely manner
which meets the changing needs of its stakeholders [2][3].

Principles behind Agile Manifesto [2]

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development.
- Deliver working software frequently.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals.
- The most effective method of conveying information within a team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential

Car Research App - Agile Process usage - Iteration 1

- **Concept**
  - This App helps a buyer of a new or old car to evaluate all the options through a preliminary research by providing information in a concise way.
  - Car MSRP, True cost to own over five years, engine specs, and ratings from Edmunds are displayed.
  - The favorite cars can be saved for future reference and also shared with friends via email.
  - Google maps can be used to locate the dealers in a zip code on a map.
- **Requirement Gathering**
  - Requirements were gathered from experience and by talking to friends.
  - Existing Apps in the market were studied so that new functionality can be provided.
  - Got an approximate idea for the User Interface and what contents are required in the App
  - An API from the http://developer.edmunds.com/ website was required to get information about cars. The user asks a query and the App makes a call to the API. Data is obtained in raw format as a JSON file which should be processed and displayed in a user friendly way.
- **Requirement Review**
  - The Edmunds API was studied to understand the queries and their content.
  - Registered on Edmunds website to get access to API keys.
  - Once the information that can be obtained from the API was known, the App feature list was updated to complete the work in available time.

Iteration 1

- **Design**
  - A GUI mock up was made for the requirements and reviewed for User friendliness from the customer perspective.
  - A data flow diagram was developed.
  - Once the Interface design seemed appropriate, the component level design was developed in a Top down manner using pseudo code.
  - The Agile software development process notes was reviewed from the class.
- **Coding**
  - A Top Down coding process was followed with a goal in mind over multiple phases.
  - The first step was to install Android SDK (software development kit) and Eclipse IDE (Integrated Development Environment) from the website developer.android.com.
  - A sample program was run to understand the IDE and Android framework.
  - Calling Edmunds API from Android program.
  - Parsing the raw JSON data from the API and understanding how to present it to the user in a meaningful way.
  - The process was first worked out for one query and Testing was part of this process.
  - Coding was done in a modular way so that some parts can be re-used later.

Iteration 1

- **Testing**
  - Testing was done in phases as each part of the App was developed.
  - Stubs were used in place of final methods for initial testing in the spirit of Top Down design.
  - Iterative testing was done as code was developed in each phase.
  - Black box testing of the interface was performed to access if everything is self explanatory.
  - White box testing of functionality was done. Each button has a specific task, and that was tested.
  - The Android Emulator was used for testing the App at this stage.
- **Customer Review**
  - In this stage, the App progress was reviewed from developer and customer perspective and future plans were made for refining the UI and the data presentation.

Iteration 2

- **Requirement Review**
  - Reviewed all the required queries from Edmunds API documentation and shortlisted the ones most appropriate for this App.
- **Design**
  - Listed Input parameters required by the API to run the queries and parsed the required results from those queries.
  - For certain results, more than one query had to be run.
  - GUI refactored to make the layout better.
- **Coding**
  - After the process was formalized for one query in iteration 1, the remaining queries were also built and parsed in a similar way.
  - Stepwise refinement was performed to present the results in a readable manner
  - Code modularity was followed again and some parts were re-used from iteration-1.
- **Testing**
  - Once all the GUI buttons and queries were programmed, complete system tests were performed and bugs were fixed.
  - The App was tested on a Nexus 7 tablet and updates were noted.

Iteration 3

- **Design**
  - Fine tune the GUI style and add buttons for Home, Save, Delete, Email etc.
- **Coding**
  - Added code for the Home, Save, Delete and Maps buttons.
  - Realized that getting Google maps to work is a lot of technical effort.
- **Testing**
  - Testing the entire App with latest features and updates.
  - Making sure the features work as expected and removing any bugs.
- **Customer Review**
  - Fixing bugs in existing functionality from user testing results.
  - Updating and adding new features to improve the utility of the App like replacing Maps with settings.
  - Reformatting the query results in a more readable way.

Iteration 4

- **Design**
  - GUI updates by replacing the Maps button with Settings and reformatted the query results.
- **Coding**
  - Removed the maps buttons and added a settings button to store info like email addresses.
  - Added code for Settings and Email buttons.
  - Formatted the query results in a user friendly way.
  - There is “Coupling” within the major buttons in the home screen where models can only be obtained after make, and year were selected.
  - The task buttons to display specific information in screen 2 like Ratings, True cost to own are “Cohesive”, i.e., independent of other tasks.
- **Testing**
  - Testing the entire App with latest features and updates.
  - Observability – the results should be relevant to the query.
  - Controllability – distinct output from each query.
  - Decomposability – independently test each button.
- **Release and User Training**
  - Release the App to the customer and conduct user training.
- **Maintenance and Support**
  - This should be an ongoing effort over the life cycle of the App.

Implementation

- The first step was to install the Android SDK (software development kit) from the developer.android.com website and the Eclipse IDE (Integrated development environment).
- Then some sample programs were run to get familiar with Eclipse environment and Android emulator.
- Based on the conceptual and design and analysis from the software process, the basic framework of the App was developed.
- 2 screens, AsyncTask to run the Edmunds API queries using Http libraries
- Processing the returned JSON results and extracting required information. This code was reused for the remaining queries with required updates.
- Displaying the results in a new screen.
- Adding the Home, Settings, Save, Delete and Email buttons
- Fine tuning the GUI layout of the App to make it intuitive for the user.
- Testing the App on the Android device.

Data Flow Diagram

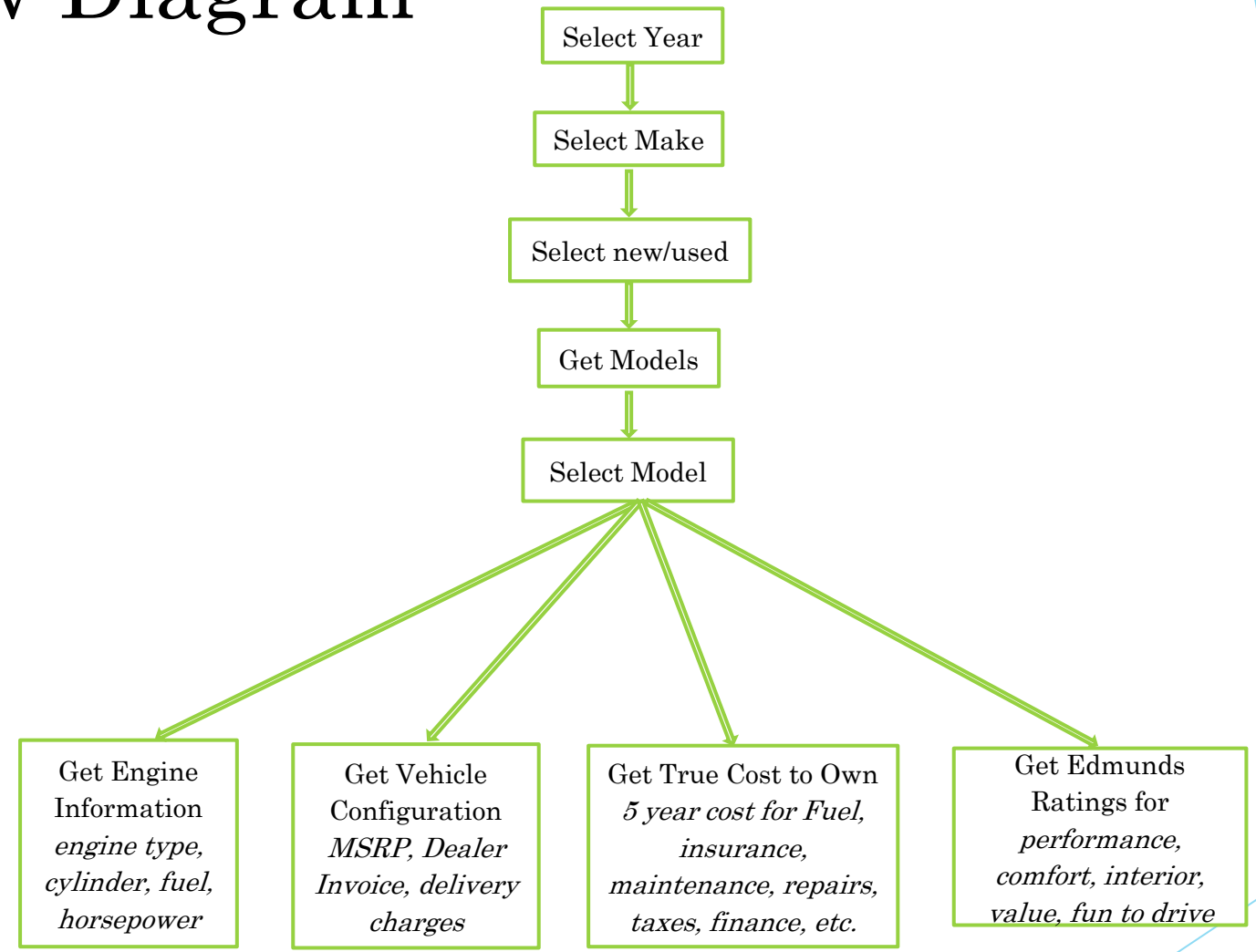


Fig 1. Data Flow Diagram

Edmunds API usage [4]

- Registered on http://developer.edmunds.com/ to get an API key.
- http://edmunds.mashery.com/io-docs webpage can be used to run queries directly on the webpage.
- **Sample query:** https://api.edmunds.com/api/vehicle/v2/mazda?state=new&year=2013&view=basic&fmt=json&api_key=zj3x5ustr7we2tz2xcwh5ty9
- This query is built from the user input as follows
  - String urlString = "https://api.edmunds.com/api/vehicle/v2/" + selMake + "?state=" + selNewOld + "&year="+ selYear + "&view=basic&fmt=json&api_key=" + apiKey;
- To run in the Android App, a HttpGet request with the urlString is sent to a DefaultHttpClient and the HttpResponse is obtained as an input stream in JSON format. This JSON data is parsed using “org.json” library. The formatted results are displayed on the screen

Observations and Learning

- Initially it seems that following a process is an overhead with so much documentation and additional work other than the real tasks.
- When many people in a team or organization are involved in collective work, a process enables better communication and level of quality which gives clarity for developers and managers and the higher management and ultimately the customers.
- It helps management to better track the progress of the project in terms of time and money.
- It makes the company less dependent on individuals and reduces risk.
- Overtime, as the company attains the highest CMM levels, it will be more efficient than the competitors in terms of product pricing, quality and ability to cope with uncertainty in business condition improves.
- The software processes stated in class can be adapted in an organization according to the specific requirements and they can be improved over time.
- As the world becomes more and more software dependent, establishing a robust software process framework is good for the employees, companies and the society.

Summary

- Using a software process helps organize the project in a meaningful way.
- A lot of redundant ideas and features can be eliminated in the requirement review phase.
- Design helps to fine tune the work to be done and creates a workable plan.
- Lot of interesting ideas can be evaluated in the iterations in the requirement, design and coding phases and only those that make sense from customer perspective can be retained after customer feedback.
- Once the Agile software development process framework and usage is established, the next project can be more efficient.

References

[1] http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

[2] http://agilemanifesto.org/principles.html

[3] http://blog.sei.cmu.edu/post.cfm/what-is-agile

[4] http://developer.edmunds.com/

QUESTIONS ?

THANK YOU !