Development of an App in Android Smart Phone for Pavement Roughness Estimation

DESIGN DOCUMENT

Team

sdmay20-17

Client

Bo Yang

Advisor

Halil Ceylan

Team Members/Roles

Tanner Dempsay
Christian Royston
Kyle Eckrich
Justin Kuennan
Joe Van Treeck
Greg Starr

Team Email

sdmay20-17@iastate.edu

Team Website

http://sdmay20-17.sd.ece.iastate.edu/

Revised

10/06/2019

Executive Summary

Development Standards & Practices Used

- Department of Transportation (DOT)
- Electrical and Computer Engineering (ECE)
- Global Positioning System (GPS)
- International Roughness Index (IRI)

Summary of Requirements

- Android application that records accelerometer data and location data during a vehicle's route
- Route data, including acceleration, GPS, etc. is sent to the server in real time
- Server calculates IRI value of pavement vehicle is driving on and transmit this back to phone
- All data is stored in a MongoDB or similar database structure
- User interface for application is easy to use while operating motor vehicle. Data is clearly visualized while the pavement is being measured

Applicable Courses from Iowa State University Curriculum

The following courses included content applicable to our project. These include topics such as Android development, project management, data communications, and user-interface design. Prerequisites taken earlier in the curriculum, such as MATH 165, will assist with the development and implementation of the roughness calculations.

- COMS 309
- CPRE 185
- PHYS 221
- CPRE 288
- MATH 165

New Skills/Knowledge acquired that was not taught in courses

Since our team is entirely made up of computer engineers, the skills we have acquired in our coursework overlap significantly. This also means that most knowledge relevant to this project that is not taught in our curriculum has to be learned by all members. Server implementations are one area we have to acquire. The requirements for the backend specifically request a NodeJS-based backend, which in itself is JavaScript-based. This language is not taught in the computer engineering core curriculum. This is the same situation for database management. Though some members have exposure to these platforms from their COM S 309 group projects, more research will need to be done to will understand these tools.

Our team is also researching the current methodology and tools for measuring pavement roughness. For this project to be useful to the end users, we need to build a model that can accurately measure the roughness using a smartphone. This background knowledge includes different types of pavement material, standards for measuring roughness, and use cases for measuring.

Table of Contents

List of figures/tables/symbols/definitions

1 Introduction

1.1 Acknowledgement

Our team would like to thank our faculty advisors Bo Yang and Halil Ceylon for allowing us to work on this project with his team and for all future assistance. We would also like to thank our Doctoral student advisor Chen-Yeou Yu and professorial advisor Wensheng Zhang in advance for helping to guide our team. We look forward to working on this project over the next year and hope to learn from our advisors and all others who helped.

1.2 Problem and Project Statement

There is a large need to monitor and characterize the overall quality of roads in order to prioritize maintenance of infrastructure. The existing solution, class 1 profilometers, are expensive systems that must be mounted to the vehicle. This prohibits small organizations from obtaining and is costly for very large organizations to maintain a fleet of such devices. Resultantly, there is a need for a cheaper solution for determining the roughness of pavement.

Our solution to this problem is a smartphone application that can calculate the International Roughness Index (IRI) of a given surface. The phone, mounted to the car, will collect the accelerometer data which will be used to calculate the IRI. Once the calculation is completed, the application will then store the determined IRI associated with the road in question, determined by the GPS of the device. As smartphones are ubiquitous, this will drastically decrease the cost of pavement monitoring.

1.3 Operational Environment

The App will be used on a phone that has been mounted to the dash of the car or truck. Since it will be indoors it won't need any special protection other than what the phone already offers. The car mount however should be very sturdy and stable as we will need to pick up the changes in gyroscope so to the phone sway unnecessarily will change the reading of the app in a negative way.

1.4 Requirements

Functional Requirements:

- Project requires an Android app representing the front end for the user, and a server for the backend.
- Android app must take accelerometer data from the device's onboard sensors.
- App must log the route taken by the user during the roughness measuring.
- Data acquired shall include GPS coordinates, accelerometer values in all axes, and IRI values among each step of the route.
- System shall use a user-based system to identify app users.
- The server shall be backed on NodeJS framework.

- All relevant data relating to routes, users, etc. will be stored in a MongoDB database system.
- Application shall have minimum API support for Android Nougat and newer versions of Android.
- Application shall be able to calculate pavement roughness offline if the cellular connection is weak or nonexistent.
- All data gathered during the route shall be saved on smartphone's local storage in case of connection error.

Economic Requirements:

- Android application shall perform optimally on most phones commercially available.

Environmental Requirements:

- Application must be able to generate accurate roughness calculations in various motorized vehicle types.

UI Requirements:

- Frontend user interface must by easy to use by the user while they are operating a motor vehicle.
- Interface should show real-time accelerometer data, location coordinates, and IRI value during route measuring.

1.5 Intended Users and Uses

This project is being designed for the future use by transportation departments within government organizations. These groups will use our product to test a large range of pavement, usually on roads, across a geographic area. In these various areas, the cell service and GPS reception quality may change and could affect the ideal functionality of the application.

The application will be run on an Android device mounted to the vehicle being driven, possibly in different orientations. The driver or passenger of the app may interface with the application during routing or use. Different vehicles could also be used to take measurements. The organization using our project will run an instance of our server application and database.

1.6 Assumptions and Limitations

Assumptions:

- Accurate IRI calculations can be performed using only accelerometer data
- The driver of the vehicle must drive a certain speed while using application
- Calculations are being performed from a vehicle driving on a pavement surface
- The smartphone being used has an accelerometer and data connection

Limitations:

- Less accurate than high cost class 1 profilometers
- Will require the user to have an Android smartphone
- May be calculation intensive, requiring server-side calculations which could potentially limit the number of concurrent users
- Cellular signal may limit the frequency of data from client to server or vice versa

1.7 Expected End Product and Deliverables

- Android application - May 2019
    - Used to collect accelerometer and GPS data while the user is driving
    - Calculates the IRI of the road driven on
- Server/Database - May 2019
    - Alternative calculation of IRI if the calculation is too intensive for a phone
    - Maintain all calculated IRI values and their associated roads

The smartphone application will simply be a tool that will allow the user to determine the roughness of a road. Before driving over a stretch of pavement, the user will attach the phone to the windshield of the vehicle, start the session, and then drive over the road. The application will then use the derived formula to generate a value to assign the roughness of the stretch of road.

There will a server and database that exist to support the application. While unknown at the moment, it is expected that the formula used to calculate the IRI value could be very time intensive. If it is, it could be too slow for the phone to perform the calculations, and we would then use the server to do these calculations. Also, the IRI values will be stored along with the road segment driven on the database to allow the user to retrieve past data.

Both tasks are intertwined with each other and require significant work to complete. As such, the deliverables are both scheduled to be completed by the month of May 2019. For a more detailed analysis of the tentative project schedule, please see Section 4.1.

2. Specifications and Analysis

2.1 Proposed Design

The proposed design is to create an Android-based application to record accelerometer data and the geographic location of the device. The device will transmit the accelerometer data and the geographic location data to a database. Both the device and database will then calculate an International Roughness Index (IRI) value using the quarter-car model. The device will then display the calculated IRI values on a map of roads. Optionally, an image can be recorded at the time of accelerometer data collection and stored along with the accelerometer and geographic location data. IRI standards must be followed by the value calculation.

None of the application or database have yet to be developed. Research into the IRI value calculation has be done and some theories on the calculation have been developed.

2.2 Design Analysis

We have met with the professor and grad students that proposed the project and talked about what the IDOT wants in their app since they are the client. We have researched about IRI calculation as well. We have a couple of test apps that were supplied to us to build from one of the professors. Showing how to get sensor read off and to store it into a file that can be analyzed for further use. As well as a test app that gathers GPS location for a path. I work well but it is just a starting point we need to do all the back end and get the calculation working.

I think one of the big strengths of the proposed solution is how little you need to get started. All you need is a smartphone something everyone has, a car mount something that is relatively cheap, and a car or truck to make the calculations. Though I would some a weakness is the calibration for all the various cars and truck we need to find a good way to adjust the calculations for different suspenders.

2.3 Development Process

We are following an Agile development process, because most team members have experience and are comfortable with the process. Using an Agile process will also allow for our team to meet checkpoints sooner, which will make sure we stay on track to finish. Lastly, using Agile will allow us to split the project into tasks which will give everyone things to work on.

2.4 Design Plan

The IRI calculation exists for the situations in which a user wants to determine the roughness of a road. The application will allow for a user to instruct the application to begin IRI calculation and to transmit the data to a central database. The application will also record the GPS location of the device in this situation. In a separate use-case, a user will want to know the roughness of roads that have already been recorded. In this situation, the user will want to access a map overlay with the recorded pavement roughnesses. The application will contain a separate user interface containing the map that will allow for a user to scroll around the map and identify the roughnesses that have already been calculated.

3. Statement of Work

3.1 Previous Work And Literature

Similar applications exist with varied amounts of implemented features. Some of the existing products have IRI calculation implemented alongside GPS tracking, which is the end goal for our product. For our product, we will not be using previous products and will be developing the app. The IRI calculation will be developed based off the accelerometer data gathered from the smartphone. Currently there are IRI equations and research about pavement roughness is available online.

3.2 Technology Considerations

The biggest strength about the technology that is currently available is that it calculates IRI accurately, so we have a basis to test our data around. Having some examples to look at also gives us an idea about what to avoid. The biggest issue with some apps is that they do not use GPS tracking with the IRI calculations, or they do not have ways to store the data. Our goal with the project is to have an application that has the required features as well as some quality of life features. Most existing products calculate IRI on the frontend side as the accelerometer data is coming in. For our project, we will have a way to calculate IRI on the smartphone, but we will also be sending data to a server and store it in a database so that it takes some load off the phone.

3.3 Task Decomposition

The development cycle for this project can be broken down into two main platforms: the frontend Android application, and the backend NodeJS server. Each platform has multiple tasks that are dependent on them, and some that require sufficient progress complete on both. This requires close communication between members working on either platform in order to complete more system-level tasks, like getting data from the database to the Android application.

1. Create NodeJS server
2. Create database with tables for data types and attributes
3. Link server and database for data reading/writing functionality
4. Develop user-interface for Android app
5. Create networking framework for frontend to communicate with backend
6. Develop IRI calculation algorithm for acceleration data from Android phones
7. Implement algorithm on backend and frontend for online and offline functionality
8. Add GPS tracking and logging to Android app
9. Couple GPS and IRI data to create path histories for users.
10. System testing with frontend, backend, and database

3.4 Possible Risks And Risk Management

Possible risks that could arise would be time management. With a proper schedule our team should be able to keep on pace to finish. Another risk would be lack of advanced knowledge on some of the programming platforms. If our team decides to implement functions that we are unfamiliar with, then issues could arise when learning how to work certain functions. Lastly risks could arise with debugging. We plan to have our product finished early so that we have time to debug the code and fix any issues that might arise.

3.5 Project Proposed Milestones and Evaluation Criteria

Milestone 1: Server and database set up and functional.

Milestone 2: User-interface for android app functional.

Milestone 3: Frontend to backend communication set up.

Milestone 4: IRI calculation in place and working on app.

Milestone 5: Online and offline functionality for calculation in place.

Milestone 6: GPS tracking functional for app.

Milestone 7: Application is able to pull and store GPS and IRI data into the database.

Milestone 8: All Android functions implemented together and functioning.

Milestone 9: Collected data is tested and all the application's features are fully implemented and functional.

Milestone 10: User-interface is set up for ease of use and quality of life. Product is finalized and looks nice.

3.6 Project Tracking Procedures

We will be making use of the Issues page on the ECE GitLab, where our Git repository for the project is located. There, we will be able to create a board similar to that of a scum-Agile project board. Requirements will be broken down into software features to be implemented, or the "issues". Issues that are in progress will be labeled as such and assigned to the team member working on that specific feature. Issues will be categorized by frontend and backend and grouped into milestones that will track the big picture progress of the project.

In addition, we will be releasing reports throughout this semester and the next on our team website. These will detail all work done on the project, including design and technical related progress being contributed by team members.

3.7 Expected Results and Validation

The desired outcome of the product is to have an android application that can accurately measure pavement roughness. The app will be mounted in a car or truck and will take the accelerometer data off the phone. The application will be easy to use and have relevant GPS and IRI data for the user to view. The final result should also have the appearance and accuracy of a finished product.

In order to test the accuracy of the application, we will take data from roads that have been measured by the Civil Engineering department. This department has an accurate device for measuring roughness that we can use to compare data. We will also test our application using a similar vehicle in order to get completely accurate data.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

This project is comprised of two major sections, the frontend and backend. To implement these components, our group will split into two smaller teams, each focusing on one of the platforms concurrently. The schedule shown below shows the tasks outlined in section 3.3 of this document with their start and end dates. The schedule shown below shows the tasks outlined in section 3.3 of this document with their start and end dates.

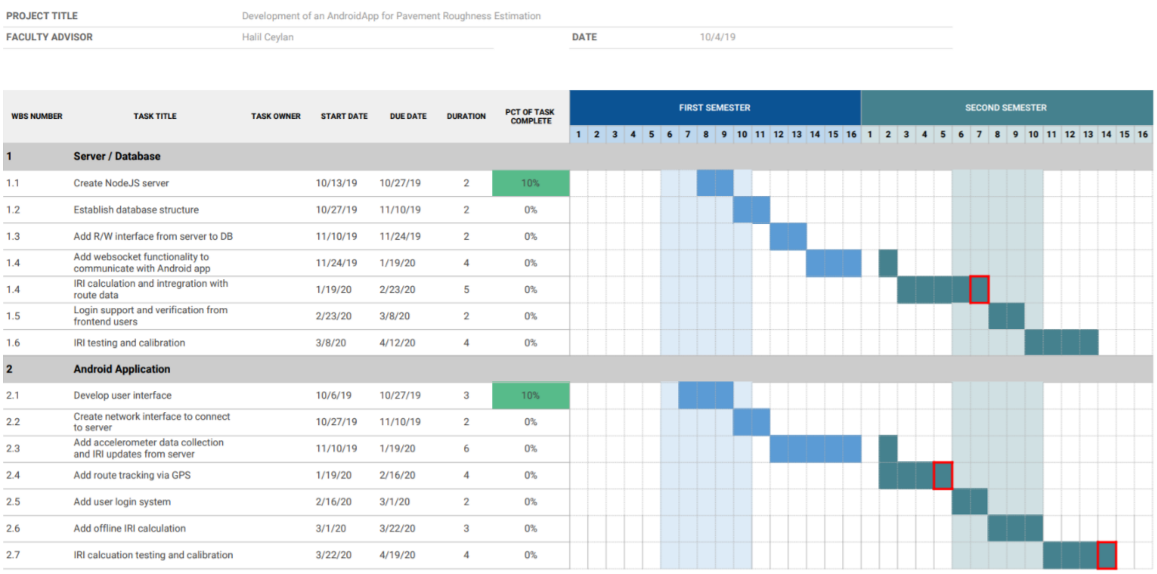

Figure 1: Project Schedule Gantt Chart

The length of each task corresponds to the estimated time to implement all of its required features. More complex tasks have longer work periods. Most of the tasks are centered around software features being implemented to each respective platform. During each "feature" task, there is a certain amount of time included in these tasks to account for basic testing of the additions, such as unit-testing.

A block outlined in red represents a deliverable or release for our project. Due to the length of development for some of our project components, we are adopting a Kanban-like release cycle, which will align with significant features working on the application and server. The server deliverable represents completion of the IRI calculation using the real-time accelerometer data from the Android application. At this point, there should be transmitting and receiving of route data and roughness measurements on the frontend, with a functional route tracking interface (first Android deliverable). The second Android release corresponds to the end of the project, where all requirements have been fulfilled and every essential feature of the frontend and backend is fully functional.

To complete this project in two semesters, we are striving to complete essential features in the first semester. Essential features are those that other aspects of the project depend on to function. For example, web socket communication between the app and server will allow the sensor data to be transmitted during routing. We will develop these using a combination of top-down and bottom-up design, where we will use our high-level requirements to create these low-level interfaces which can be adapted and reused for various features. We are also leaving substantial time for testing near the end of the project's lifecycle to test our platforms on low, high, and system levels. Since IRI measurement devices already exist, we will be able to test our calculations against existing data. We want to use this to toughly refine and prove our measurements to create product that end users will trust when they use it.

4.2 Feasibility Assessment

The challenges with this project will be the dynamic environments in which the application will be used. While the IRI calculation may be a known equation, people may use our product in many different cars with different mounting systems, suspension, etc. Rigorous testing will need to be done to ensure we are producing a consistent, accurate measurement tool for people.

Another challenge will be the routing function of this application. During the measurement cycle, the application software will have to manage location data, accelerometer data, network interactions, and local storage all in real time. During the creation of this program's structure, we will have to keep in mind the time required for each of these steps and possible multi-threaded solutions to take advantage of modern phones' hardware.

4.3 Personnel Effort Requirements

The initial few tasks of the project are mainly setting up the server and database and making sure everything is connected correctly. This will not take too much time, so a few hours for each should be enough time. The next few parts are setting up the android app and making sure it can communicate with the backend. This will take a little longer than in order to make sure everything is communicating correctly with each other. The next part is developing the actual calculation for IRI. This will take a few weeks to convert data we can pull off of the smartphone into a value that people can use. We will also be combining this with a few days of testing to make sure that the values are correct. The next task is to set up offline and online calculations, which could be tricky, so we assigned one week to figuring this out. Task 8 will be setting up GPS data. This task has been allotted three weeks so that we can get it working accurately. The next task will be making sure everything works together, which would take about a week, mainly dealing with bugs that arise. Lastly, we need to test our product, so a few days have been allotted to make sure our app is accurate.

Tasks	Estimated Time
Task 1:	2 hours
Task 2:	2 hours
Task 3:	4 hours
Task 4:	6 hours
Task 5:	10 hours
Task 6:	3 weeks
Task 7:	1 week

Task 8:	3 weeks
Task 9:	1 week
Task 10:	2 days

TABLE 1: PERSONNEL EFFORT REQUIREMENTS

4.4 OTHER RESOURCE REQUIREMENTS

- Android-based device to test the application
- Server space to hold database data
- LASER-based IRI value calculator and vehicle to compare the application values

4.5 FINANCIAL REQUIREMENTS

This project requires Android smartphones for all developers, a server, and a car to be used during the derivation and testing of the IRI calculation method, however we will be able to supply the car. As some team members already have Android smartphones, we will only need three additional devices, although it is possible that the department or someone related has an extra, unused device. Finally, we will also need a physical server to host our database and for server-side code. While the cost of this is low for a project of our size, the cost scales with the number of users, data stored, and computation time used.

Required Item	Count	Unit Cost	Total Cost
Android Phone	3	$0 (if used phones are supplied), $50+	$0-150+
Server (Development phase)	8 months	$0 (Free trials)	$0
Server (Deployment)			~ $5/month
Total			$0-150, $5/month

Table 2: Financial Requirements

5. Testing and Implementation

5.1 INTERFACE SPECIFICATIONS

For testing, an Android-based device will be required. The Android-based device will utilize the accelerometers included with the phone to calculate an International Roughness Index value using the quarter-car model. Testing will be conducted with a vehicle including a LASER-based IRI calculator. The values of the applications calculation and the LASER calculation will be compared, and a determination will be made on whether the application's calculation is acceptable.

5.2 Hardware and software

All our testing will be software testing, as there is no physical component to this project. We will be performing unit testing throughout the entirety of the development of the application, to ensure all the low-level code works exactly as intended. As we build the application from disparate pieces of code, we will then perform integration testing to ensure, for example, that the accelerometer values collected are successfully transferred to the server. Additionally, as we develop our algorithm for calculating the IRI value, we will need to be continuously testing our program in comparison to known values. Finally, we will perform user acceptance testing to ensure all functional and non-functional requirements have been adequately met.

5.3 Functional Testing

Although not limited to these, we will mainly be performing unit and integration. Unit testing is an integral part of code development, as the programmer should be aware if the functions written operate as intended. We will do this frequently in our development cycle, but examples include testing the accrual of accelerometer and GPS data and accuracy of the IRI calculations.

Integration testing becomes useful as the codebase of the project increases in complexity. When code written at different times is brought together, it must be ensured that there are no unexpected behaviors produced. Again, this is common, but examples in our project could include testing the interaction between the collection of GPS data its association with the calculated IRI value in our database.

5.4 Non-Functional Testing

Our project has very few non-functional requirements, many of which are simply properties of the application that cannot be tested, such as the requirement that it be an Android application. The only major requirement we may test is in relation to the user interface and ease of use. To test this we will need to, safely, use our application while driving a vehicle while monitoring how distracted we are and determine how easy it is to use.

5.5 Process (Not Done)

The IRI calculation will be tested by comparing the calculation with a separate calculation conducted by a LASER-based method. This testing will determine whether the differences in the LASER-based calculation and the Android-based calculation are within a percentage of error or not. Communication between the server and the device will be conducted through testing whether information on the server can be received by the device. Map information will be tested by comparing information stored on the database with information that the device receives from the database.

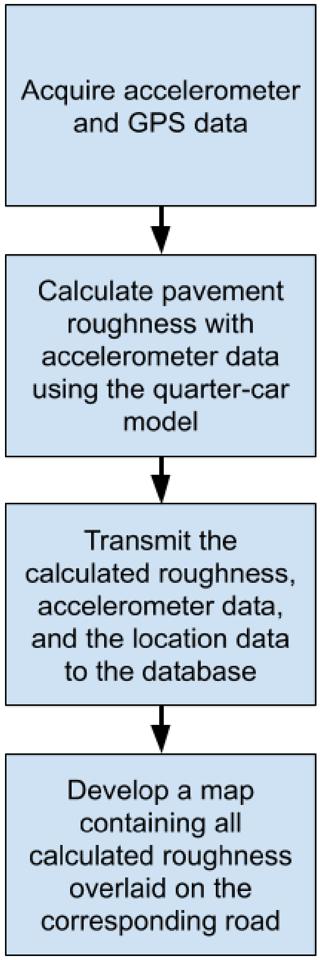

Figure 2: Process Diagram

5.6 Results

No testing has yet to be conducted. Thus, there are yet to be any results.

6. Closing Material

6.1 Conclusion

The team has researched International Roughness Index value calculations and determined tasks that each member will complete. The goal of the project is to create an Android application that will calculate the IRI value of a portion of road using accelerometer data and map the IRI calculations of roads. The most likely way of achieving these goals is so create an Android application utilizing the accelerometers in an Android-based device and the GPS of the device. The collected data will be stored on a database. These methods will allow for the data to be collected quickly and stored sufficiently for future reference.

6.2 References

The team does not have any references at this moment.

6.3 Appendices

At the current version of the project, there are no relevant appendices that we are referencing.