Mobile Augmented Reality Number Plate Detection Android System

Project Management Information

Compiled by *The Gruners*

Department of Defense, Peace, Safety and Security, at the
Council for Scientific and Industrial Research (DPSS, CSIR).

Table of Contents

Background

This document contains all Project Management information pertaining to the Mobile Augmented Reality Number Plate Recognition System to disclose how the whole implementation of the system (from requirements design through development phase) was embarked upon by The Gruners team.

Software Development Process

For one to develop an efficient software system, one must use a certain framework that will help them plan, develop and control practical issues that may rise during the development process of the system.

Development Method

Due to the fact that our system was research based, most of the technologies that were used had never been used for the purpose of this kind of software system. Consequently, most of the requirements kept changing and the scope was always fluctuating.

For this reason, we chose Agile Methodologies as our development method so as to make it easy for us to react well to continuously changing requirements.

This development method was applied in the following ways:

- The first phase was building a System Requirements Specification document. This documents specifies the vision, scope, features, interfaces and other attributes of the envisioned system. It also shows the priorities of the different requirements to help us make early and continuous delivery of valuable functionalities.

- Secondly, we added into the requirements document an architecture specification. This part makes solid guidelines on the foundation on which the system will be built. This decision of the architectures to use was made initially based on experience of technologies and certain techniques. However, this architecture changed vastly during the development phase and we introduced specifications that would help address quality requirements of the system.

- Thirdly came the development phase. This part of the process kept iterating numerous times as we ran tests and reviewed back on our requirements and architecture specifications. This part of the development is the main reason why which chose this development method.

- Lastly we built manual and automated tests for our system. These tests were done per requirement after implementation of each requirement so that the particular requirement can be reviewed against the requirements specification document in order to make changes that would more efficiently address functional and/or non-functional requirements of the system.

Team Members and Responsibility Assignment

Below is a table that articulates all team members, their profile, as well as their responsibilities. For a more detailed profile for the team members, see the Team Portfolio document.

Names	Brief Profile	Responsibilities Fulfilled
Ndivhuwo Nthambeleni (Inactive)	**Skills** - C++, Java, C#, JQUERY, MySQL - HTML5, CSS, PHP, JavaScript, XML - Android Development **Work Experience** - Tutor (University of PTA) - Intern (CSIR)	- System Requirements Document. - Remote Database Design
Joas Mogale	**Skills** - C#, C++, Java, VB, Delphi, Python. - MySQL, MS SQL Server, PostGreSQL, DB4O, BaseX, SQLite3 - HTML, CSS, PHP, JavaScript, XML, JQuery, JSon, Ajax, Ruby on Rails, Django. - Photoshop, Blender. **Work Experience** - Web Designer (JayPee Media)	- System Requirements Document. - Web Interface - Graphics Design - Optical Character Recognition - Remote Database Design - Android Database Design - Manual Integration Testing
Mbulungo Justin Musetsho (Leader)	**Skills** - C#, C/C++, Java, VB, Python. - MySQL, MS SQL Server, PostGreSQL, DB4O, BaseX, SQLite3 - HTML, CSS, PHP, JavaScript, XML, JQuery, JSon, Ajax, Django. - Photoshop. **Work Experience** - Tutor (University of Pretoria) - Teaching Assistant (University of Pretoria)	- System Requirements Document. - REST Web Service - Business Logic - Android REST Client - Optical Character Recognition - Remote Database Design - Manual Integration Testing

Issue Management

A goal of this process is to pinpoint issues as soon as we discover them and to resolve them diligently. The issue management process was never handled by a single member, but rather by all team members. This was to raise practical issues either inside or outside of their areas of specific responsibility when they become aware of them.

The following steps were followed in the process of managing practical issues

- Verify, confirm and record an identified issue.
- Collaboratively inform each other about the issue.
- Agree on a date on which the issue must be resolved.
- If the issue affects progress of other processes, then continue working with the processes using mock data where possible.
- If resolution is not found amongst team members by the agreed upon due date, notify lecturers and client
- Once the issue is solved, integrate with the involved processes and perform tests to confirm solution.
  .

Project Status VS Time

For each demo, a table is shown elaborating all the tasks and roles that we had and their progress thereof.

Demo 1

No.	Task	Not Started	In Progress	Completed
1	Requirements Doc			X
2	Architecture Spec			X
3	Project Timeline			X
4	Android Interface	X		
5	Web Interface	X		
6	Web Service	X		
7	Business Logic	X		
8	Android REST Client	X		
9	Audit Log	X		
10	Number Plate Detection	X		
11	Optical Character Recognition	X		
12	Neural Adjustments	X		
13	Get Detection Location	X		
14	Web Logic	X		
15	Detect Different Number Plate Dimensions	X		
16	Unit and Integration Test	X		
17	Detection History	X		
18	Performance Fulfillment	X		
19	Accuracy Fulfillment	X		
20	Detect Vehicle Features			

Demo 2

No.	Task	Not Started	In Progress	Completed
1	Requirements Doc			X
2	Architecture Spec			X
3	Project Timeline			X
4	Android Interface		X	
5	Web Interface			X
6	Web Service			X
7	Business Logic	X		
8	Android REST Client	X		
9	Audit Log	X		
10	Number Plate Detection	X		
11	Optical Character Recognition	X		
12	Neural Adjustments	X		
13	Get Detection Location			X
14	Web Logic		X	
15	Detect Different Number Plate Dimensions	X		
16	Unit and Integration Test	X		
17	Detection History	X		
18	Performance Fulfillment	X		
19	Accuracy Fulfillment	X		
20	Detect Vehicle Features	X		

Demo 3

No.	Task	Not Started	In Progress	Completed
1	Requirements Doc			X
2	Architecture Spec			X
3	Project Timeline			X
4	Android Interface			X
5	Web Interface			X
6	Web Service			X
7	Business Logic		X	
8	Android REST Client		X	
9	Audit Log	X		
10	Number Plate Detection	X		
11	Optical Character Recognition	X		
12	Neural Adjustments	X		
13	Get Detection Location			X
14	Web Logic			X
15	Detect Different Number Plate Dimensions	X		
16	Unit and Integration Test	X		
17	Detection History	X		
18	Performance Fulfillment	X		
19	Accuracy Fulfillment	X		
20	Detect Vehicle Features	X		

Demo 4

No.	Task	Not Started	In Progress	Completed
1	Requirements Doc			X
2	Architecture Spec			X
3	Project Timeline			X
4	Android Interface			X
5	Web Interface			X
6	Web Service			X
7	Business Logic			X
8	Android REST Client			X
9	Audit Log	X		
10	Number Plate Detection		X	
11	Optical Character Recognition		X	
12	Neural Adjustments	X		
13	Get Detection Location			X
14	Web Logic			X
15	Detect Different Number Plate Dimensions	X		
16	Unit and Integration Test	X		
17	Detection History	X		
18	Performance Fulfillment		X	
19	Accuracy Fulfillment	X		
20	Detect Vehicle Features	X		

Demo 5

No.	Task	Not Started	In Progress	Completed
1	Requirements Doc			X
2	Architecture Spec			X
3	Project Timeline			X
4	Android Interface			X
5	Web Interface			X
6	Web Service			X
7	Business Logic			X
8	Android REST Client			X
9	Audit Log		X	
10	Number Plate Detection			X
11	Optical Character Recognition			X
12	Neural Adjustments			X
13	Get Detection Location			X
14	Web Logic			X
15	Detect Different Number Plate Dimensions		X	
16	Unit and Integration Test		X	
17	Detection History			X
18	Performance Fulfillment			X
19	Accuracy Fulfillment		X	
20	Detect Vehicle Features		X	

The following diagram shows the current status of the project as well as all the functionality that was implemented and when they were completed.

Project Status VS Demo Days
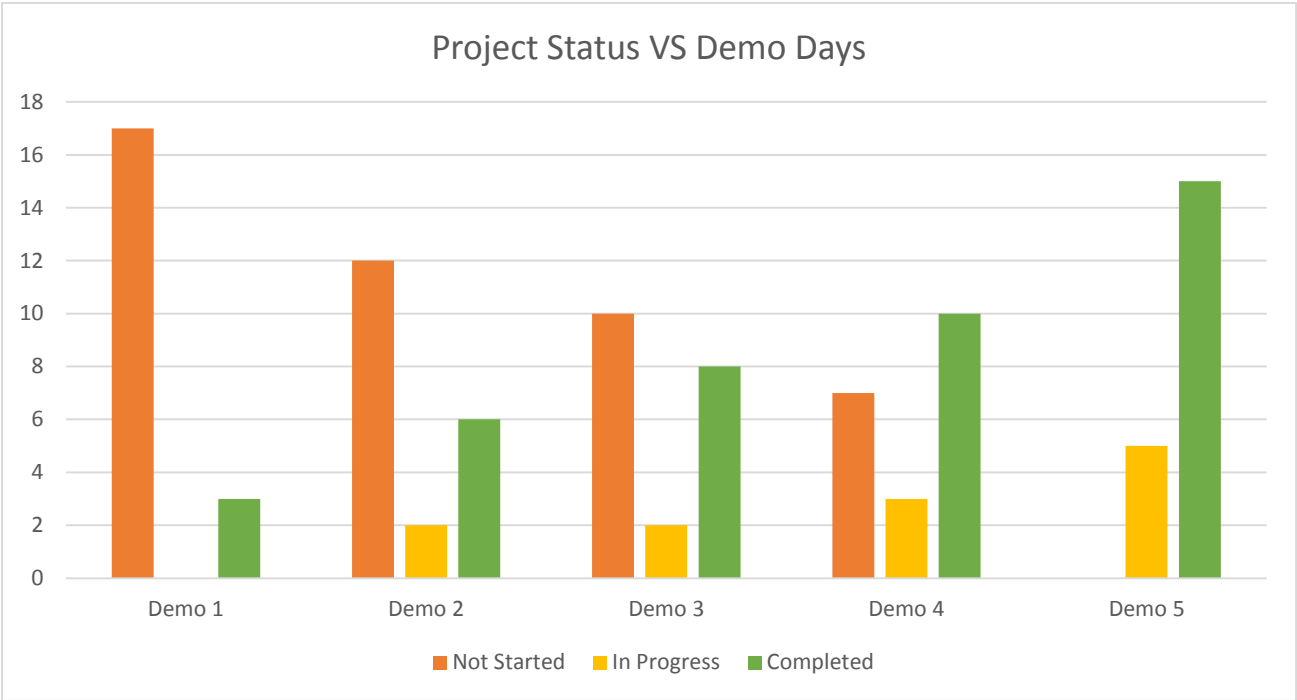
Unimplemented Functionality

At this stage there is not a single functionality that has not been started. However there are some functionalities that we could not finish implementing.

Below is an abstract list of unimplemented functionality:

- Audit Log
- Detect Different Number Plate Dimensions
- Unit and Integration Tests
- Accuracy Fulfillment
- Detect Vehicle Features (Color, Make, Brand, Type)

Main Challenges and Risks

Almost every phase of the development process came with its issue, most of which were caused mainly by the fact that the technologies specified by the client did not fulfill the perceived aim. Below is a list of challenges that we came across.

SDK Irrelevance

When the development phase of the Android system started, we picked up an issue with the Optical Character Recognition SDK. This SDK only allows detection of words that are in a particular dictionary, hence making the use of this SDK inappropriate for this purpose. Having discovered this problem, we decided to implement all the functionality dependent on Optical Character Recognition with mock data during the time when we were investigating on a solution for this issue.

Number Plate Differences

For our algorithm that detects the number plate of a car within a picture, we had difficulties finding enough sample number plates to train the application. This had a large impact on the fulfillment of the performance and accuracy quality requirements.

Member Inactivity

Immediately after the first phase of the project, one member of the team became inactive. At this time he had roles and responsibilities to fulfill and we unfortunately only found out after 6 weeks. This brought a lot of lagging into the project development process. Consequently, some of the implementations took time to complete (if completed at all).