



Project Thinmint

Requirement Specifications

Girl Scout Cookies:

Bobby Cafazzo

Chad Crumley

Mike McHugh

Jason Torosian

Cory Zinkiewicz

Table of Contents

1. System Context & Scope	1
1.1. Scope of the System	1
2. Functional and Data Requirements.....	1
2.1. Functional Requirements.....	1
2.2. Data Requirements	1
3. Look, Feel and Use Requirements.....	1
3.1. User Interface Requirements	1
3.2. Style of the Product Requirements	2
4. Performance Requirements.....	2
4.1. Reliability and Availability Requirements	2
4.2. Capacity and Scalability Requirements.....	2
5. Operational Requirements	2
5.1. Expected Technological Environment Requirements	2
5.2. Supportability, Maintainability and Portability Requirements	2
6. Security Requirements.....	3
7. Off-the-Shelf Solutions Requirements	3
8. Constraints and Assumptions	3
9. Future Requirements.....	3
10. Appendix.....	4
Figure 1 – GUI Mockup.....	4
Figure 2 – Use Case Diagram	4

1. System Context & Scope

1.1. Scope of the System

The purpose of Project Thinmint is to record and display the speed of a standard wobbler engine and the coinciding pressure. The team will be able to calculate the rotations per minute of the engine using a Netduino board and Magnetic sensor. That input data, along with the input pressure, can then be manipulated via coding and then produce useful output data. The program will display to the user the relationship between the current RPM of the engine as well as the supplied pressure.

2. Functional and Data Requirements

2.1. Functional Requirements

Project Thinmint's hardware includes a Wobbler Engine, an air compressor, a PSI sensor and a magnetic input sensor. The air compressor allows the user to provide an exact amount of air pressure to the engine. A magnetic input sensor will provide the computer program with the current RPM of the engine.

Project Thinmint's program component will feature several diagnostic tools to aid in the analysis of the Wobbler Engine's performance. These features include current RPM and PSI displays; an RPM and PSI predictor, which will allow the user to input a target RPM to achieve; several graphing tools; and a graphical representation of the Wobbler Engine's speed.

2.2. Data Requirements

Project Thinmint will require no permanent storage to perform its diagnostic operations. All data read in and out will only be stored in memory temporarily.

3. Look, Feel and Use Requirements

3.1. User Interface Requirements

The user interface will display the current RPM and PSI, as well as a graph showing the correlation between the two. There will also be input fields for the two values that will allow the user to predict the required input.

See Figure 1 – GUI Mockup for a mock-up of the user interface.

3.2. Style of the Product Requirements

The wobbler engine is carved out of T6061 brushed aluminum and mounted to a stylized baseplate. The graphical user interface will be designed to match this. The graphical representation of the Wobbler Engine will also change based on its performance, where engines that have got to go fast will clearly look like they're going fast.

4. Performance Requirements

4.1. Reliability and Availability Requirements

Because Project Thinmint requires an air compressor, the system can only be run in the vicinity of one. In this case, the air compressors available to us are wall-mounted in Wentworth's multiple laboratories; actual usage scenarios must take place in said laboratories.

Wentworth Wobbler Engines are already known for their durability so there will be no issues with hardware replacement, only maintenance in the sense that it will need to be periodically oiled. Because the software side will not have any network connectivity or permanent storage, there will be no opportunity for data loss as well.

4.2. Capacity and Scalability Requirements

The diagnostic tool of Project Thinmint will be able to run diagnostics on any type of engine which it can get input from. In this sense, the tool has near infinite scalability when it comes to engine analysis.

5. Operational Requirements

5.1. Expected Technological Environment Requirements

Because Project Thinmint requires an air compressor, the system can only be run in the vicinity of one. In this case, because the air compressors available to us are wall-mounted in Wentworth's multiple laboratories, actual usage scenarios must take place in said laboratories.

5.2. Supportability, Maintainability and Portability Requirements

The wobbler engine must maintain adequate lubrication to ensure minimal friction when spinning. The air compressor must also be able to maintain constant PSI to maintain adequate readings, because of this the wall-mounted air compressors will be ideal for the project, meaning that Project Thinmint will require a very specific laboratory environment to run in.

6. Security Requirements

Because there is no network connectivity or permanent data storage in Project Thinmint, security is not a major concern for this system.

7. Off-the-Shelf Solutions Requirements

Off-the-shelf solutions for Project Thinmint include...

- **Air Compressor** – a heavy-duty air compressor capable of exerting at least 60 PSI constantly over a long period of time.
- **Magnetic Input Sensor** – a small magnetic sensor able to detect every time the Wobbler's wheel has completed a full revolution and forward that information to the Netudino.
- **PSI Sensor** – a pressure sensor able to read the amount of air being pushed into the system and forward that information to the Netudino.

8. Constraints and Assumptions

The product has the following requirements: the air compressor must be able to exert at least 60 PSI constantly over an extended period of time. The computer software must be run on a Windows-based operating system running the .NET Framework at version 4.0 or later. It is also non-intrusive and does not require a great deal of processing power. Sensors must be used within their functionality range.

9. Future Requirements

Future extensions of Project Thinmint may include a PSI regulator for exact control of RPM.

10. Appendix

Figure 1 – GUI Mockup

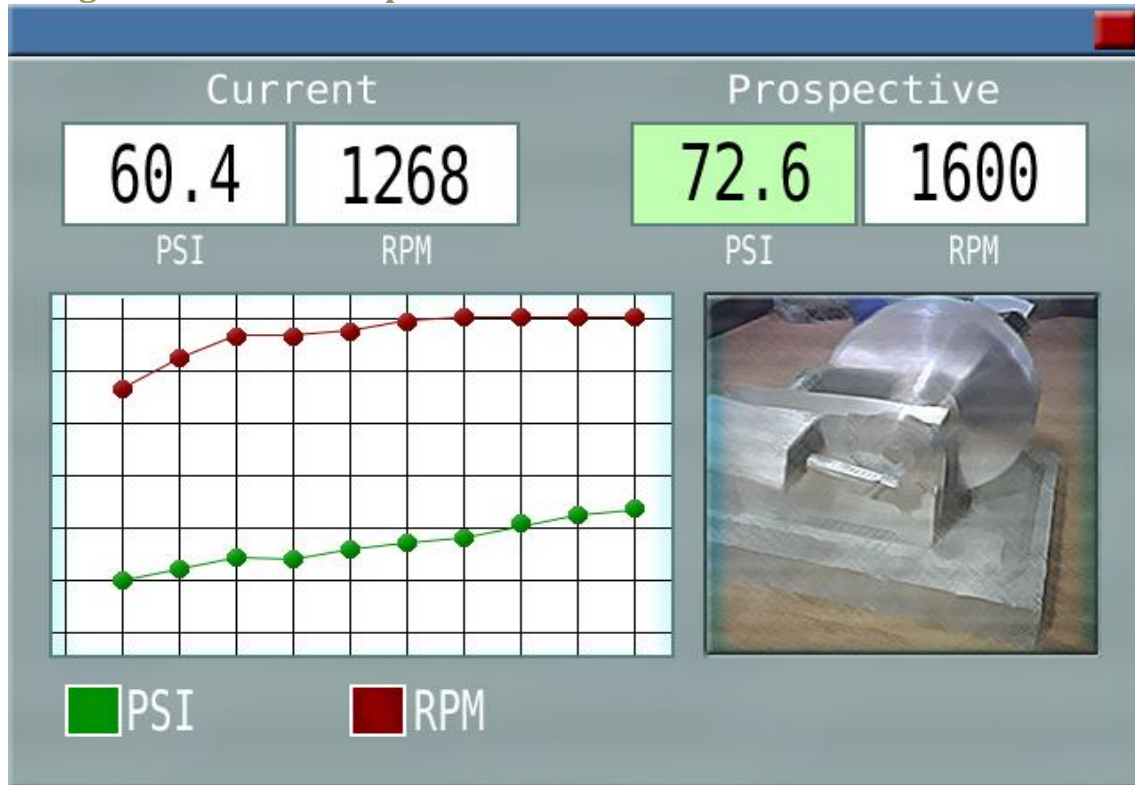


Figure 2 – Use Case Diagram

