Functional Specification

1 Introduction

1.1 Product Overview

The product is designed to help small engineering businesses keep track of all the maintenance and repair tasks for their machines.
The idea is that when a machine breaks down, the operator of the machine logs into their operator account and marks their machine as inactive, after which the operator is then asked to write a small description of what the reason was behind the machines shutdown. The engineer will then receive the notification on their engineer account as well as the description of the problem.
The engineer ideally should have an idea of the problem and know what to expect before they enter the factory floor. Once there, the engineer will identify the source of the problem and then use the app to check if a similar problem has occurred in the past and how a previous engineer has fixed it. The engineer should now have the required knowledge to fix the problem, saving valuable time by providing the solution immediately.
The purpose of this project is to provide assistance to engineers making documentation less tedious and more easily accessible, thus speeding up the process for maintenance engineers doing their jobs.

1.2 Project Scope

By the time of project completion the project should be able to accomplish the following,

- Have a working login system for three types of users (admin, engineer, operator).
- Allow the engineers to record information about their repairs, calibrations and diagnostics for each individual in house.
- Record dates and times for audits calibrations and diagnostics and inform the user when they are due
- The operator account should be able to mark a machine as inactive and notify the engineer user of why said machine is inactive
- The admin account should be able to create new accounts for each type of user.
- The project must be able to display a list of machines and within those machine lists each machine must have a list of parts attached
- If possible a search bar function should be implemented for specific machines and specific parts
- A tutorial video for the user will be accessible from the front page
- The engineer will be able to add repair notes to each individual part as well as the machine itself
- A possible statistics feature could be implemented recording breakdown rates of each individual machine informing the user when the rate of breakdown rises above a specific threshold

- The operators account should have a visual layout of each machines location to help with identification of the machine in the room in which they operate (a map of the machines that represents the room layout for example)
- If the engineer has encountered a problem that has occurred before the engineer should just be able to mark the problem as a recurring problem along with the date on which it occurred this saves writing up the same problem multiple times

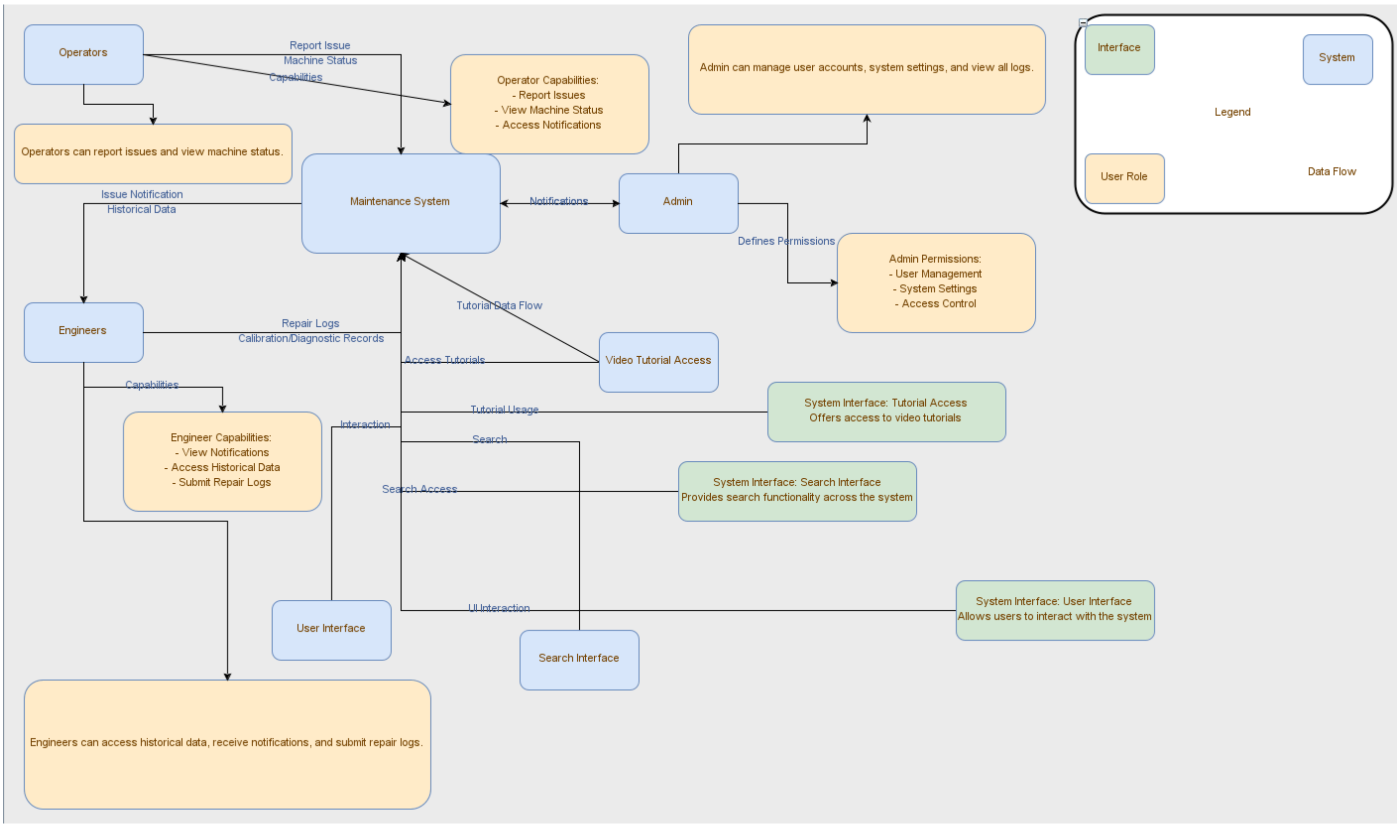
  (will add more features later if we can realistically implement them) - James

2 Solution Overview

The product will be written in Java and will be provided as a desktop application, and the possibility of an additional web-app for portability is to be considered as well.
The application will access a cloud database so that it is not restricted to being used in one location (providing you have internet access), however, it also enables users to store files temporarily offline until they have a network connection and can then sync any data.

(Currently updating and working on - William)

2.1 Context Diagram



2.2 System Actors

2.2.1 User Roles and Responsibilities

User/Role	Example	Frequency of use	Features used
Operator	Individual machine operator	Occasional, only when there is a machine fault	Minimal Access, logging machine faults and receiving confirmation when fixed
Engineer	Machine engineer	Frequent, notified when maintenance is required	Receiving maintenance requests, machine part lists, maintenance video watching. Making part purchase requests
Admin	System/company administrator	Occasional	Setting up accounts. Viewing machine statistics. Receiving part purchase requests

2.3 Dependencies and Change Impacts

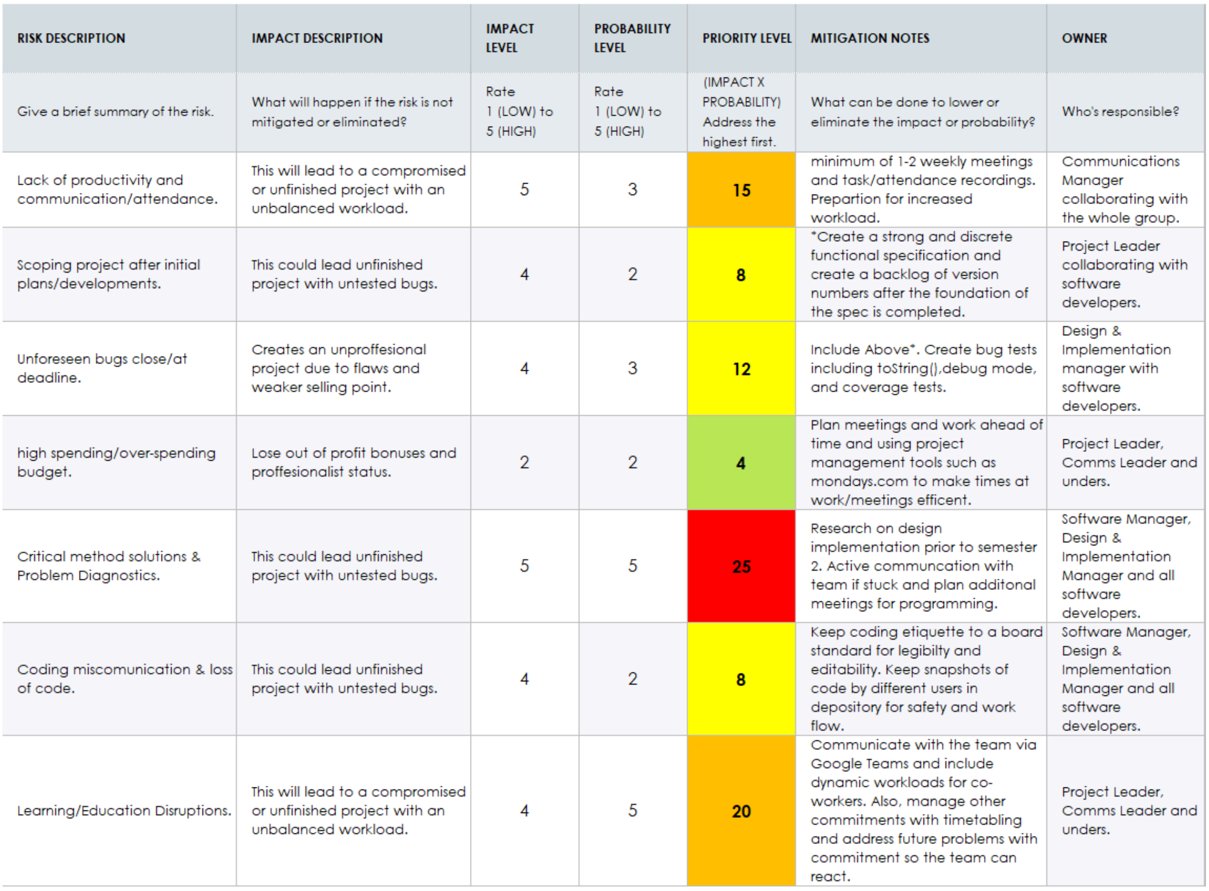
2.3.1 System Dependencies

- Access to the internet for a cloud server or a local server to store data such as machine status, logs, login details, images and videos.
- Computers that are capable of running the software and are compatible with Java.

2.3.2 Change Impacts

- How issues with machines are reported.
- How maintenance/repair is recorded.
- How machine statistics are handled.
- How administration manages the engineers and operators.

2.4 Risks and Mitigations



RISK DESCRIPTION	IMPACT DESCRIPTION	IMPACT LEVEL	PROBABILITY LEVEL	PRIORITY LEVEL	MITIGATION NOTES	OWNER
Give a brief summary of the risk.	What will happen if the risk is not mitigated or eliminated?	Rate 1 (LOW) to 5 (HIGH)	Rate 1 (LOW) to 5 (HIGH)	(IMPACT X PROBABILITY) Address the highest first.	What can be done to lower or eliminate the impact or probability?	Who's responsible?
Lack of productivity and communication/attendance.	This will lead to a compromised or unfinished project with an unbalanced workload.	5	3	**15**	minimum of 1-2 weekly meetings and task/attendance recordings. Prepartion for increased workload.	Communications Manager collaborating with the whole group.
Scoping project after initial plans/developments.	This could lead unfinished project with untested bugs.	4	2	**8**	*Create a strong and discrete functional specification and create a backlog of version numbers after the foundation of the spec is completed.	Project Leader collaborating with software developers.
Unforeseen bugs close/at deadline.	Creates an unproffesional project due to flaws and weaker selling point.	4	3	**12**	Include Above*. Create bug tests including toString(),debug mode, and coverage tests.	Design & Implementation manager with software developers.
high spending/over-spending budget.	Lose out of profit bonuses and proffesionalist status.	2	2	**4**	Plan meetings and work ahead of time and using project management tools such as mondays.com to make times at work/meetings efficent.	Project Leader, Comms Leader and unders.
Critical method solutions & Problem Diagnostics.	This could lead unfinished project with untested bugs.	5	5	**25**	Research on design implementation prior to semester 2. Active communcation with team if stuck and plan additonal meetings for programming.	Software Manager, Design & Implementation Manager and all software developers.
Coding miscomunication & loss of code.	This could lead unfinished project with untested bugs.	4	2	**8**	Keep coding etiquette to a board standard for legibilty and editability. Keep snapshots of code by different users in depository for safety and work flow.	Software Manager, Design & Implementation Manager and all software developers.
Learning/Education Disruptions.	This will lead to a compromised or unfinished project with an unbalanced workload.	4	5	**20**	Communicate with the team via Google Teams and include dynamic workloads for co-workers. Also, manage other commitments with timetabling and address future problems with commitment so the team can react.	Project Leader, Comms Leader and unders.